CHAPTER 8

SUPPORTING INTERNET APPLICATIONS OVER THE ATM-ABR SERVICE

With the proliference of multimedia traffic over the Internet, it seems natural to move over to ATM technology which has been designed specifically to support integration of data, voice, and video applications. While multimedia applications are still in the development stage, most of the traffic on the Internet today is data traffic in the sense that they are bursty and relatively delay insensitive. It is, therefore, natural to ask how the current applications will perform over the ATM technology.

Although ATM technology has been designed to provide an end-to-end transport level service and so, strictly speaking, there is no need to have TCP or IP if the entire path from source to destination is an ATM path. However, in the forseeable future, this scenario is going to be rare. A more common scenario would be where only part of the path is ATM. In this case, TCP is needed to provide the end-to-end transport functions (like flow control, retransmission, ordered delivery) and ATM networks are used simply as "bit pipes" or "bitways."

Since the Available Bit Rate (ABR) and the Unspecified Bit Rate (UBR) service classes have been developed specifically to support data applications, it is important to investigate the performance of dominant internet applications like file transfer and

world wide web (which use TCP/IP) running over ABR and UBR. In this dissertation, we concentrate on the performance of TCP/IP over ATM-ABR. ATM-UBR performance has been examined in several recent studies [32, 31, 89, 38]. The aforementioned TCP studies also compare UBR performance with ABR using either EFCI switches [31] 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. In this chapter, we provide a more detailed study of the dynamics and performance of TCP over ABR.

In the UBR service class, the only degree of freedom to control traffic is through scheduling, buffer allocation and cell drop policies. ABR has additional degrees of freedom in terms of switch schemes and source parameters. The ABR service requires network switches to constantly monitor their load and feed the information back to the sources, which in turn dynamically adjust their input into the network. The Transport Control Protocol (TCP), at the same time, uses packet loss in the subnetwork as an implicit feedback indicating network congestion and reduces its data load on the network. This mechanism is called the "Slow Start" congestion avoidance mechanism [45]. There is currently a debate in the networking community about the need for ABR service particularly in light of TCP's built-in congestion control facilities. We address some of these issues in this chapter.

We first study the dynamics of TCP traffic over ATM, the effect of cell loss, and the interaction of TCP with the ABR congestion control mechanisms. We find that TCP performs best when it does not experience packet loss. For the ABR service, we quantify the amount of buffering required at the ATM switches to avoid TCP packet loss. Specifically, we find that ABR is scalable over TCP in the sense that it requires

buffering which does not depend upon the number of connections. The amount of buffering depends upon factors such as the switch congestion control scheme used, and the maximum round trip time (RTT) of all virtual circuits (VCs) through the link. On the other hand, the UBR service is not scalable in the sense that it requires buffering proportional to the sum of the TCP receiver windows of all sources.

The above observations are true for applications like file transfer which have persistant demand characteristics. We verify that the requirements hold even in the presence of highly VBR background traffic (including multiplexed MPEG-2 video traffic). However, when TCP applications are bursty (i.e., have active and idle periods), it is possible that the network is overloaded by a burst of data from a number of TCP sources simultaneously. While there can be little guarantees under such pathological workloads, we find that our observations about buffer requirements hold for a large number of World Web Web (real-life bursty) applications running over TCP.

8.1 TCP control 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 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 8.1.

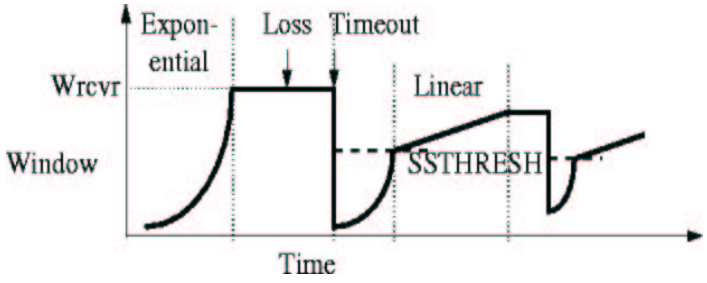


Figure 8.1: TCP Window vs Time using Slow Start

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 8.2. 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.

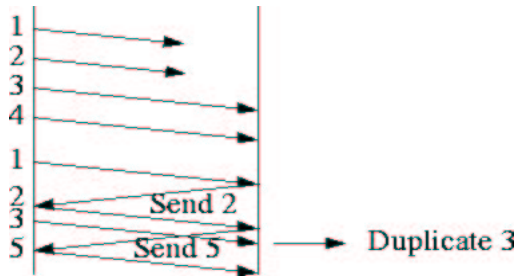


Figure 8.2: Timeout and Duplicate Packets in Slow Start

8.2 Closed Loop vs Open Loop Control Revisited

The ABR service provides flow control at the ATM level itself. 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). The network feedback is effective after a time delay. The time delay required for the new feedback to take effect is the sum of the time taken for an RM cell to reach the source from the switch and the time for a cell (sent at the new rate) to reach the switch from the source. This time delay is called the "feedback delay."

When the source transmits data after an idle period, there is no reliable feedback from the network. For one round trip time (time taken by a cell to travel from the source to the destination and back), the source rates are primarily controlled by the ABR source end system rules (open-loop control). The open-loop control is replaced by the closed-loop control once the control loop is established. When the traffic on ABR is "bursty" i.e., the traffic consists of busy and idle periods, open-loop control may be exercised at the beginning of every active period (burst). Hence, the source rules assume considerable importance in ABR flow control.

8.3 Nature of TCP Traffic at the ATM Layer

Data which uses TCP is controlled first by the TCP "slow start" procedure before it appears as traffic to the ATM layer. Suppose we have a large file transfer running on top of TCP. When the file transfer begins, TCP sets its congestion window (CWND) to one. The congestion window increases exponentially with time. Specifically, the window increases by one for every ack received. Over any round trip time (RTT), the congestion window doubles in size. From the switch's point of view, there are two packets input in the next cycle for every packet transmitted in the current cycle (a cycle at a bottleneck is defined as the largest round trip time of any VC going through the bottleneck). In other words, the load (measured over a cycle) at most doubles every cycle. In other words, ***initially, the TCP load increases exponentially.***

Though the application on top of TCP is a persistant application (file-transfer), as shown in Figure 8.3, *the TCP traffic as seen at the ATM layer is bursty (i.e., has active and idle periods).* Initially, there is a short active period (the first packet is sent) followed by a long idle period (nearly one round-trip time, waiting for an

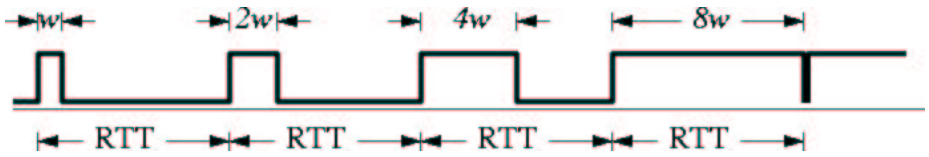


Figure 8.3: At the ATM layer, the TCP traffic results in bursts. The burst size doubles every round trip until the traffic becomes continuous.

ACK). The length of the active period doubles every round-trip time and the idle period reduces correspondingly. Finally, the active period occupies the entire round-trip time and there is no idle period. After this point, the TCP traffic appears as an infinite (or persistant) traffic stream at the ATM layer. Note that the total TCP load still keeps increasing unless the sources are controlled. This is because, for every packet transmitted, some TCP source window increases by one which results in the transmission of two packets in the next cycle. However, since the total number of packets transmitted in a cycle is limited by the delay-bandwidth product, ***the TCP window increases linearly after the bottleneck is fully loaded.*** Note that the maximum load, assuming sufficient bottleneck capacity, is the sum of all the TCP receiver windows, each sent at link rate.

When sufficient load is not experienced at the ABR switches, the switch algorithms typically allocate high rates to the sources. This is likely to be the case when a new TCP connection starts sending data. The file transfer data is bottlenecked by the TCP congestion window size and not by the ABR source rate. In this state, we say that the TCP sources are window-limited.

The TCP active periods double every round trip time and eventually load the switches and appear as infinite traffic at the ATM layer. The switches now give

feedback asking sources to reduce their rates. The TCP congestion window is now large and is increasing. Hence, it will send data at rate greater than the source's sending rate. The file transfer data is bottlenecked by the ABR source rate and not by the TCP congestion window size. In this state, we say that the TCP sources are ***rate-limited***. Observe that UBR cannot rate-limit TCP sources and would need to buffer the entire TCP load inside the network.

The ABR queues at the switches start increasing when the TCP idle times are not sufficient to clear the queues built up during the TCP active times. The queues may increase until the ABR source rates converge to optimum values. Once the TCP sources are rate-limited and the rates converge to optimum values, the lengths of the ABR queues at the switch will start decreasing. The queues now move over to the source end-system (outside the ATM network). Several proprietary techniques can be used to control the TCP queues at the edge of the ATM network [80], and we cover some of the possibilities later in this chapter.

The remaining part of the chapter is organized as follows. We first examine the performance of TCP under lossy conditions in section 8.4. We then study the interaction of the TCP and ABR congestion control algorithms and the justification and assumptions for buffer requirements for zero loss in section 8.14. Next, we look at the effect of variation in capacity (VBR backgrounds) on the buffer requirements. We discuss switch algorithm issues and our solutions to handle variation in demand and capacity in section 8.16. We then develop a model of multiplexed MPEG-2 sources over VBR, and study its effect on TCP sources running over ABR in section 8.17. Finally, we look at related work (an extension of this dissertation work), where the

issues and effect of bursty applications like World Wide Web running on TCP is examined.

8.4 TCP Performance With Cell Loss

Cell loss will occur in the network if the ATM switches do not have sufficient buffers to accomodate this queue buildup. In this section, we will show simulations to demonstrate the problem of cell loss on TCP performance and identify the factors which affect the performance under such conditions. Specifically, we show that TCP achieves peak throughput over ABR without the necessity of very large buffers (we quantify this requirement in section 8.14). Then we limit the buffer size based on the Transient Buffer Exposure (TBE) ABR SES parameter and the number of TCP sources. Though the TBE parameter was initially intended to allow some control over buffer allocation, we find that it is ineffective in preventing cell loss. We then study the effect of cell loss on TCP level packet throughput, and the various parameters affecting the performance (buffer size, number of sources, TCP timer granularity parameter, cell drop policy etc).

Specifically, when cell loss does occur, the cell loss ratio (CLR) metric, which quantifies cell loss, is a poor indicator of loss in TCP throughput. This is because TCP loses time (through timeouts) rather than cells (cell loss). Smaller TCP timer granularity (which controls timeout durations) can help improve throughput. Due to fragmentation, a single cell loss results in a packet loss. This further obscures the meaning of the CLR metric. If the ABR rates do not converge to optimum values before the cell loss occurs, the effect of the switch congestion scheme may be

dominated by factors such as the drop policy and TCP timer granularity. Intelligent drop policies can help improve the throughput slightly.

8.5 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 $\times$ 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" used in popular TCP Reno implementation. In a related work [39, 40] (TCP over UBR), we study the effect of these algorithms in detail. Briefly, these algorithms have been designed to improve TCP performance when a single (isolated) segment is lost (due to errors). However, in high bandwidth links, network congestion results in several dropped segments (a burst loss). In this case, these algorithms are not able to recover

from the loss and they trigger the TCP timeout and the slow start algorithm leading to possibly worse performance.

8.6 ABR Source End System and ERICA Parameters

The source end system parameters 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 $\times$ n $\times$ 1, 2, or 4, where n is the number of ABR sources.

8.7 The n Source + VBR Configuration

Figure 8.4 illustrates the general configuration we analyze, which we call "the n Source + VBR configuration." This configuration has a single bottleneck link

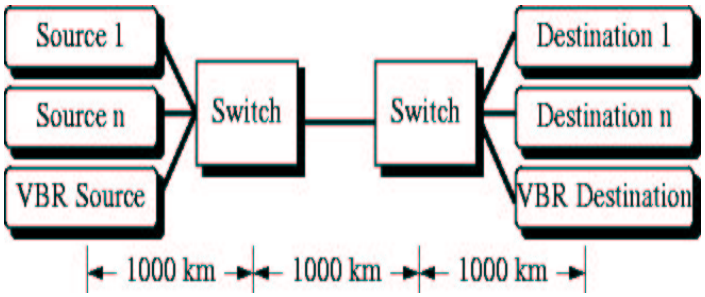


Figure 8.4: n Source + VBR Configuration

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}.

8.8 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 8.5.

8.9 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

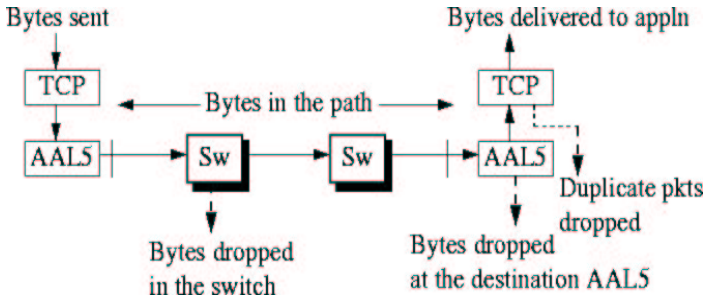


Figure 8.5: Cell/Packet Drop Points on a TCP/ATM connection

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

$\times$ 0.9 for ERICA Target Utilization

$\times$ 48/53 for ATM payload

$\times$ 512/568 for protocol headers

(20 TCP + 20 IP + 8 RFC1577 + 8 AAL5 = 56 bytes)

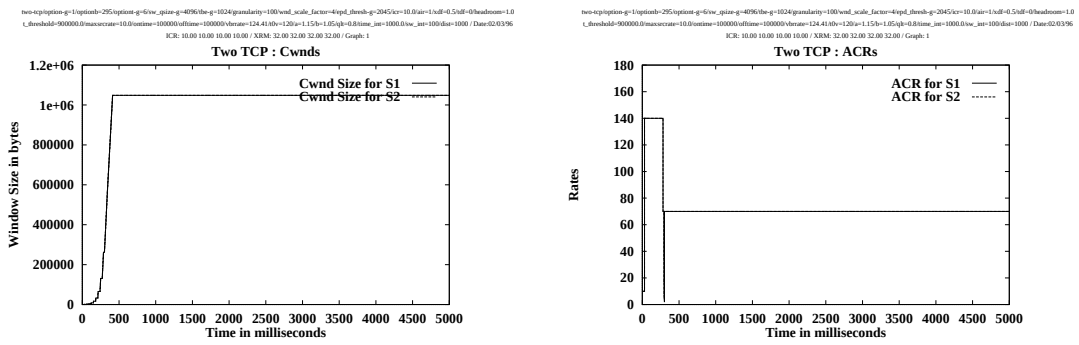
$\times$ 31/32 for ABR RM cell overhead

$\times$ a fraction (0.9) to account for the TCP startup time

$\simeq$ 103.32 Mbps

Figure 8.6 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.* Note that given sufficient time, the ABR switch algorithm can control the rates of the VCs carrying TCP traffic. We shall quantify this time and corresponding buffer requirements in section 8.14 later in this chapter.



(a) Congestion Window (b) ACR

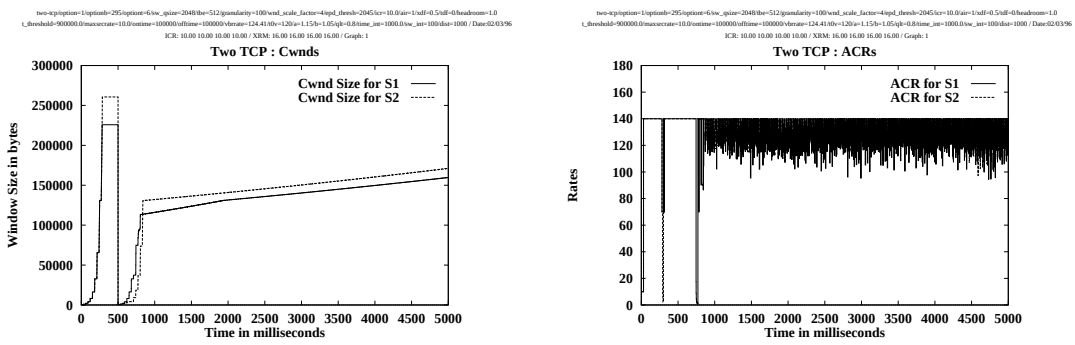
Figure 8.6: Two TCP Source Configuration, Buffer=4096 cells, TBE=1024

8.10 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., $2048 = 512 \times 2 \times 2$. The remaining configuration is the same as in Section 8.9 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 8.7 shows graphs of window size, sequence numbers, and ACR for the two sources. Figure 8.7(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.*



(a) Congestion Window (b) ACR

Figure 8.7: Two TCP Source Configuration, Buffer=2048 cells, TBE=512

Figure 8.7(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.

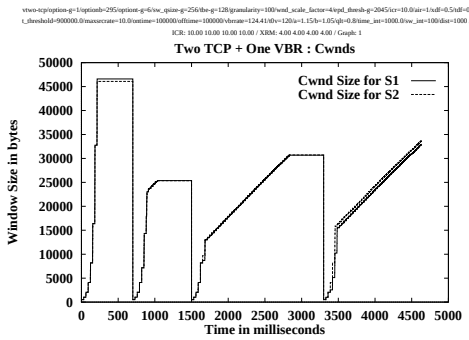
The main reason for the large drop in throughput is that cells (packets) are dropped. Each cell loss results in a significant loss of time and throughput. In this case, this happens before the ABR control loop is set up (open-loop period). The TBE in this case was 512. For two sources, one would assume 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 [70, 30].

8.11 Effect of Finite Buffers and Varying ABR Capacity

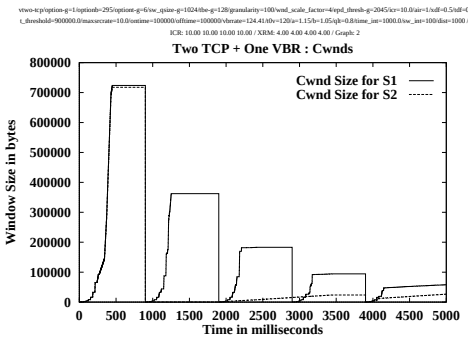
Next we studied the effect of varying ABR capacity. For this purpose, we introduce a VBR traffic in the background. We conducted several experiments with two and five ABR sources. Since VBR takes up 40% of the link bandwidth, we expect the maximum ABR throughput to be 60% of the case without VBR.

Figures 8.8 and 8.9 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

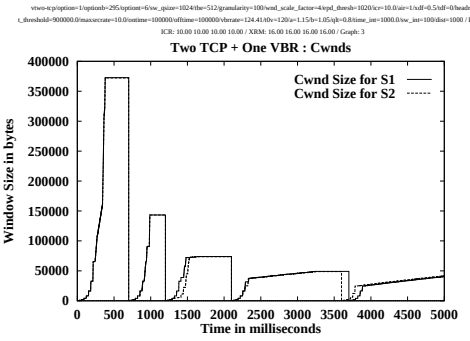
adjusted. The ACR and sequence number graphs have not been included here since there is not much new information in them.



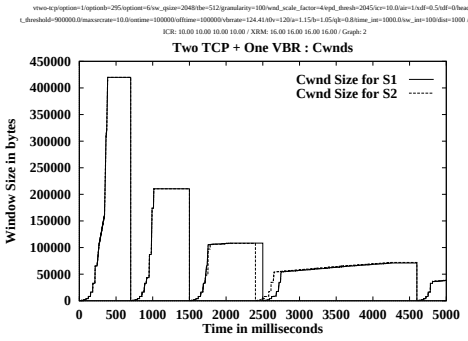
(a)
buffer=256, TBE=128, Congestion Window



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



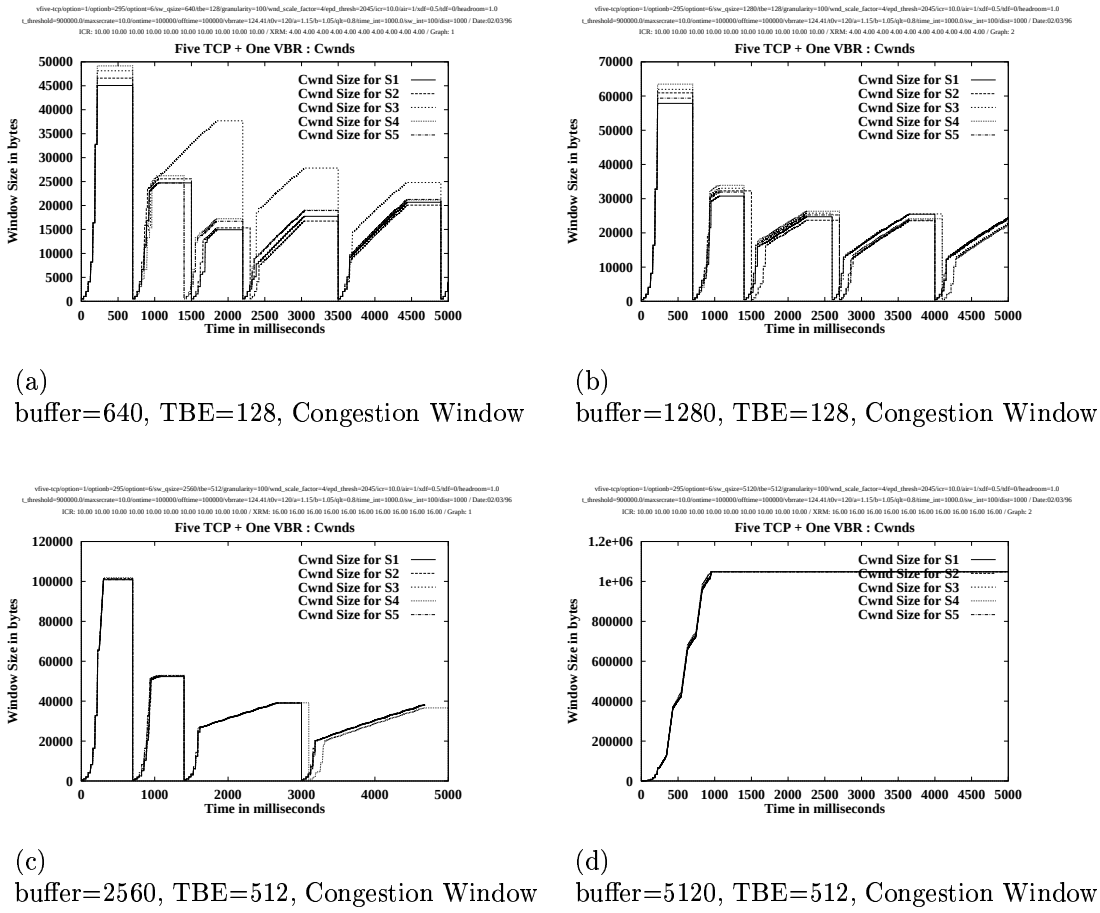
(c)
buffer=1024, TBE=512, Congestion Window



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

Figure 8.8: Two TCP + One VBR Configuration, TBE vs Buffer

The simulation results are summarized in Table 8.1 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



(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 8.9: Five TCP + One VBR Configuration, TBE vs Buffer

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

Table 8.1: Simulation Results: Summary

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.*

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

   We study the effect of buffers on latency in section 8.14. Briefly, since the ABR control can drain out the queues after the initial transient, the average latency should be low. The effect of new TCP sources starting up does not increase the latency significantly since they start at a window of one MSS, irrespective of their ICRs.

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.

8.12 Observations on Tail Drop

In this section we report an interesting phenomenon due to tail drop and propose a simple fix. In AAL5, sources mark 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 8.10. Figure 8.8(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.

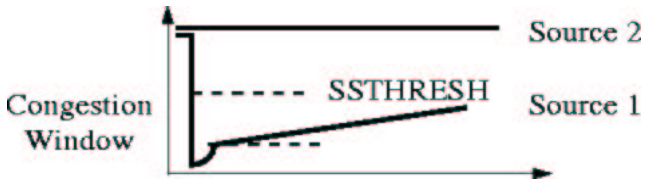


Figure 8.10: Unfairness due to TailDrop

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 first EOM cell.* The EOM cell will reach the destination and result in the dropping of the first packet and *merging of packets is avoided in the destination AAL5.* Whenever any cells are dropped, the switch should ensure that the next EOM cell is transmitted. 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) [108]. A similar policy for partial packet discard is described in Reference [5]. Drop policies assume importance for the UBR service class and have been studied in a related work [39].

8.13 Summary of TCP/IP performance over ABR under lossy conditions

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. We will quantify this requirement, and argue the case for scalability in the next section.

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 can be 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. Dimensioning should be based upon the performance of the switch algorithm, and the round trip time, as discussed in the next section.

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.

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. When sufficient buffers are provided for ABR, the drop policy assumes only a minor importance, unlike its role in the UBR service.

8.14 Buffering Requirements for TCP over ABR

In this section, we analyze the buffer requirement at switches for TCP over the ATM-ABR service. We show by a combination of emperical and analytical studies that the buffer requirement for TCP over ABR for zero loss transmission is:
($a \times$ RTT $+ b \times$ Averaging Interval Length $+ c \times$ feedback delay) $\times$ link bandwidth, for low values of the coefficients
This requirement is heavily dependent on the switch algorithm. With the ERICA+ algorithm, typical conservative values of the coefficients are ($a = 3, b = 1, c = 1$).

The formula is a linear relation on three key factors:

Round trip time (RTT): Twice the delay through the ABR network or segment (delimited by VS/VD switch(es)).

Averaging Interval Length: A quantity which captures the measurement aspects of a switch congestion control algorithm. Typical measured quantities are: ABR capacity, average queue length, ABR input rate, number of active sources, and VC's rate.

Feedback delay: Twice the delay from the bottleneck to the ABR source (or virtual source). Feedback delay is the minimum time for switch feedback to be effective.

Note that the formula does not depend upon the number of TCP sources. This fact implies that ABR can support TCP (data) applications in a scalable fashion. The buffer requirement is also an indication of the maximum delay through the network. Note that this is a worst case requirement and the average delay is much smaller due the congestion avoidance mechanisms at the ATM layer. As a result, ABR is a better fit for scalable support of interactive applications which involve data large transfers (like web-based downloading etc).

8.14.1 Assumptions

In the above formula, we have assumed that the traffic using TCP is a persistant kind of traffic (like a large file transfer). Note that it is possible for TCP to keep its window open for a while and not send data. In the worst case, if a number of TCP sources keep increasing their TCP windows slowly (during underload), and then synchronize to send data, the queue seen at the switch is the sum of the TCP windows [116].

Variation in ABR demand and capacity affects the feedback given by the switch algorithm. If the switch algorithm is highly sensitive to variation, the switch queues may never be bounded since, on the average, the rates are never controlled. The buffer requirement above assumes that the switch algorithm can tolerate variation in ABR capacity and demand. We discuss this issue further in section 8.16.

Also, in the above formula, we are assuming that the product of the number of active TCP sources times the maximum segment size (MSS) is small compared to the

buffer requirement derived. We are also assuming that the applications running on top of TCP are persistant (large file transfer type applications). Also note that the buffer requirement is for the switches only. In other words, the queues are pushed by ABR to the edged of the network, and the edge routers need to use proprietary mechanisms to manage the edge queues. We shall address these assumptions, and their implications at the end of the section.

Note also that, under certain extreme conditions (like large RTT of satellite networks) some of the factors (RTT, feedback delay, Averaging interval) may dominate over the others (eg: the feedback delay over the round trip time in satellite networks). Another scenario is a LAN where the averaging interval dominates over both RTT and feedback delay. The round trip time for a ABR segment (delimited by VS/VD switches) is twice the maximum one-way delay within the segment, and not the end-to-end delay of any ABR connection passing through the segment. These factors further reduce the buffer requirements in LAN switches interfacing to large networks, or LAN switches which have connections passing through segmented WANs.

8.14.2 Derivation of the buffer requirement

1. **Initial TCP behavior:** TCP load doubles every RTT initially when the bottleneck is not loaded (see also section 8.3). During this phase, TCP sources are window-limited, i.e., their data transmission is bottlenecked by their congestion window sizes and not by the network directed rate.

   Initially all the TCP sources are in their exponential rise phase. In its exponential rise phase, TCP doubles its window every RTT. For every cell output

on a link, two cells are expected as input to the link in the next RTT. This is true irrespective of the number of sources.

This can be viewed in three ways:

- the number of cells in the network (in an RTT) doubles
- the overload measured over an RTT doubles
- the "active period" of the burst doubles every RTT

2. **Time to reach rate-limited operation**: The minimum number of RTTs required to reach rate-limited operation decreases as the logarithm of the number of sources. In other words, the more the number of sources, the faster they all reach rate-limited operation. Rate-limited operation occurs when the TCP sources are constrained by the network directed ABR rate rather than their congestion window sizes. The minimum number of RTTs required is derived as follows:

   Suppose we find that find that TCP packets are available, but the source is not transmitting. There are two reasons for this: either the source has exhausted its window, or it is waiting for the next transmission opportunity at the current ACR. In the first case, we call the source *window-limited*. In the second case, it is *rate-limited*.

   Initially, the window is small (starting from one). The sources are window-limited (and not rate-limited). That is, each source exhausts its window and may remain idle for a while before it sends its next burst. As observed, the number of cells in the network doubles every RTT. Stable closed loop rate-control can be established only after there are enough cells to fill the pipe. The

number of cells in the first RTT depends upon the number of sources starting up together (one MSS for each source). Henceforth, the number of cells only double irrespective of the number of sources. Hence, the number of RTTs required to fill the pipe depends upon the number of sources as follows:

After K RTTs,

window $W = 2^{K-1} \times MSS$

Note that MSS = 512 bytes + Overhead = 12 cells

Or,

Effective rate = $MSS \times 2^{K-1} \times N/RTT$

Here, N = number of sources

The pipe is filled when the effective input rate first exceeds capacity (or overload becomes greater than 1, for ERICA)

$MSS \times 2^{K-1} \times N/RTT \geq L$

Here, L is the link rate in cells per second.

or

$K - 1 = \log_2 (\frac{Link\_Bandwidth \times RTT}{(MSS * N)})$

This shows that the number of RTTs decreases as the log of N. Note that once the link (or path/pipe) becomes fully loaded, the TCP windows increase linearly and not exponentially (see section 8.3 earlier in this chapter). However, the sources may still not be rate-limited (the ACRs are still large, though the bottleneck load is greater than unity). The sources reach rate-limited steady

state only after the switch algorithm brings the bottleneck load down to unity again by reallocating the rates.

3. **Switch algorithm issues:** We have claimed that the switch algorithm cannot, in general, give correct feedback when the network load is bursty (i.e., has active and idle periods). Some of problems observed by common switch algorithms are discussed below:

Figure 8.11: Out-of-phase Effect (TCP over ABR)

a) Out-of-phase effect: No load or sources are seen in the forward direction while sources and RM cells are seen in the reverse direction (figure 8.11).

Figure 8.12: Clustering Effect (TCP over ABR)

b) Clustering effect: The cells from TCP connections typically come in clusters [119] (figure 8.12). Hence, the activity of multiple connections is difficult to sense over small averaging intervals, though the corresponding load may be high.

c) **Variation in load:** As described in section 8.3, even an infinite traffic source running on top of TCP looks like a bursty source at the ATM layer. When a number of such sources aggregate, the load experienced at the switch can be highly variant.

d) **Variation in capacity:** The ABR capacity depends upon the link bandwidth, and the bandwidth usage of the higher priority classes like CBR and VBR, and can exhibit variation accordingly.

Due to these effects, switches may make errors in measuring quantities which they use to calculate feedback. Due to the out-of-phase effect, the input rate and overload measured in ERICA over the last interval is zero, because no cells are seen in the forward direction. Due to the clustering effect, the number of active sources may be underestimated in any interval (for example, N different sources may be seen in each interval when the total number of sources is 2N), leading to overallocation of rates in ERICA.

The third problem is variation in load. Due to the variation in load, it is possible to have a long period of underload, followed by a sudden burst which builds queues. As a result the maximum queue may be large even though the utilization/throughput is low. Schemes like ERICA can track the variation in load and filter it, because we use the average load as a metric. However, several schemes use the queue length metric exclusively. Queue length has a higher variation than the average load, and it also varies depending upon the available capacity. Further, a queue length of zero yields little information about the utilization of the link. It has been argued that schemes which look at only the

queue length is less susceptible to errors than schemes which use several metrics (like input rate, MACR , number of active sources etc). But, the use of several independent metrics gives more complete information about the system [66], and variation reduction can be done by using simple averaging techniques.

The fourth problem is the effect of ABR capacity variation. This effect, when combined with the latency in giving feedback to sources, results in an alternating series of high and low rate allocations by the switch. If the average total allocation exceeds the average capacity, this could result in unbounded queueing delays.

These effects reduce as the network path gets completely filled by TCP traffic, and the ABR closed loop control becomes effective. The switch scheme then controls the rate of the sources. Note that we have designed averaging techniques in ERICA (see chapter 6) specifically to counter such conditions, i.e., reduce the error in measurement and handle boundary cases. The residual error even after these modifications manifests as queues at the bottleneck. We handle this using the queue control algorithm in ERICA which scales down the ABR capacity as a function of queueing delay.

4. **Switch algorithm convergence time:** After the pipe just becomes full (TCP has sent data continuously for one RTT), the maximum queue which can build up before the switch algorithm convergences to steady state $2 \times$ RTT $\times$ link bandwidth.

   Note that the first feedback after the link is fully loaded can take as much as a round trip time to be effective. Also note (from the discussing in section 8.3)

that if not controlled, the TCP load increases at most linearly after the bottleneck is loaded, even though the sources are in their exponential rise phase. In other words, the load may change in the following pattern: cycle 0, load = 0.25; cycle 1, load = 0.5; cycle 2, load = 1, cycle 3, load = 2, cycle 4, load = 3, and so on. Note that the load change from 0.25 to 1 was exponential, and then became linear because of the bottleneck becoming fully loaded. However, even the linear load increase can result in unbounded switch queues unless the sources are controlled. The switch algorithm is therefore the key which decides how much queues build up before the sources rates stabilize. The above statements assume that the TCP windows increase smoothly and not in bursts (i.e., TCP acknowledgements are not bunched together on receipt at the sources).

With ERICA, once the link is fully loaded, the measurements (input rate, number of active source etc) can be made more reliably. There is also a continuous flow of BRM cells in the reverse direction, which can carry the rate feedback to the sources. This results in accurate feedback to sources. The sources are asked to reduce their rates, and the effect is seen at the switch within one feedback delay. Recall that the feedback delay is defined as the sum of the delay from the switch to the source and from the source to the same switch. Feedback delay is always less than a round-trip time. In the worst case, the feedback delay is equal to the round trip time.

From our observation that the overload increases at most by a factor of two every round-trip time, the maximum overload with ERICA in the first RTT after the link is fully loaded is two. The typical convergence time of ERICA (time to reach the steady state) is about two round trip times (one round trip

to equalize rates for fairness, and one round trip to bring the load to unity). See the performance evaluation of ERICA in chapter 6 for further details. Assuming that the rate allocations do not result in the increase of the total load, the link is overloaded by a factor of two during the convergence period.

The queue growth during this period is $2 \times RTT \times Link\_Bandwidth$ (the rate of queue growth is at most $Link\_Bandwidth$, since the maximum overload factor is two). However, in general, the switch algorithm takes a few round trips and a few feedback delays before it converges. The feedback delay becomes a factor because many switch algorithms give feedback as the RM cell travels in the reverse direction (rather than in the forward direction), which results in a sources responding faster to feedback. Also note that we have used an explicit rate scheme (ERICA) which results in faster convergence and hence smaller buffer requirements. A binary (EFCI) feedback scheme or an ER-scheme which is sensitive to variation may require larger buffers.

After this convergence phase, the source rates are controlled. In other words, the sources transition from a "window-limited" state to a "rate-limited" state. This is a stable state. The switch queues built up during the convergence phase are now drained out and the system reaches its "congestion avoidance" operating point of "high utilization and low queueing delay."

Note that even if a new VC carrying TCP traffic starts transmitting data, the additional load is small. This is because the TCP source sends only one segment (due to the slow start algorithm which starts with a window of one) irrespective of the Initial Cell Rate (ICR) of the VC. Note that switches have to provide additional buffering for sources which may carry non-TCP sources and may

start transmitting at ICR. ICR negotiation is therefore important for buffer allocation. Further, we also assume that the RTTs are sufficiently larger than the MSS, so that the addition of one segment of size MSS from every new source starting (at a window of one) does not increase the queue size significantly. Another assumption, is that the applications which run on TCP are either persistant, or have sufficient idle time between transmissions to allow resetting of TCP windows . The latter condition is because TCP implementations reset their congestion windows to one if there is no source activity for more than a TCP timeout period (typically a few hundreds of milliseconds) [118].

5. **Queue Backlogs:** In the above analysis, we have ignored the queue backlog due to bursts smaller than RTT. This backlog is built up as follows:

   The TCP sources have active periods and idle periods. The idle periods are used by the switch to clear out the backlogs created during the active periods. In general, TCP creates an temporary overload of 2 during the active period (which is less than RTT initially). This active period is followed by an idle (zero load) period for $RTT - activeperiod$ which is used to clear the backlogs created during the active period. We will assume that the switch algorithm is totally ineffective as long as the active period is smaller than RTT. The active period doubles every RTT until it is greater than RTT. Once the active period is greater than RTT, the switch rate control takes over.

   Suppose idle period is T. The corresponding active period is $RTT - T$. The maximum queue buildup during this active period is:

$$QB = Link\_bandwidth \times (RTT - T)$$

and the maximum queue drain during the idle period is:

$$QD = Link\_bandwidth \times T$$

Observe that $QD > QB$ when $T > RTT - T$, i.e., when the idle period $T > RTT/2$

In other words, the backlogs build up only when the idle periods become smaller than $RTT/2$. From the nature of TCP traffic, we know that initially the idle times are large, and then they reduce in size exponentially. When the idle time becomes smaller than $RTT/2$, the maximum queue backlog in each cycle (till the link becomes fully loaded) is $Link\_Bandwidth \times (RTT - 2T)$. But, observe that such a backlog can be created only once. This is because a burst needs to have an idle time T which satisfies $0 < T < RTT/2$ in order to create a backlog. Since, the active period doubles every RTT, there cannot be two RTTs where the idle time T satisfies $0 < T < RTT/2$.

6. **Effect of two-way traffic:** The above analysis has assumed undirectional TCP traffic (typical of file-transfer applications). We will briefly study the effect of two-way traffic on the buffer requirements. It has been noted [119] that bidirectional traffic complicated TCP dynamics considerably leading to more bursty behavior by TCP. This is called the "Ack Compression" phenomenon.

TCP smoothes out its window increase by using small steps on the receipt of every new ack from the destination. If the flow of acks is smooth, the window increases are smooth. As a result, TCP data cannot load the network in sudden bursts. However, when the traffic flows in both directions on TCP connections, the behavior is more complex. This is because the acknowledgements may be received in bursts, rather than being spaced out over time. As a result, the TCP window increases by amounts larger than one MSS, and the source sends data in larger bursts.

In the worst case, all the acknowledgements for the previous cycle are received at once, i.e., acknowledgements for $D = 1 \times \text{RTT} \times \text{Link\_bandwidth}$ bytes of data is received at once. This results in the $2D$ bytes of data (D due to data using the portion of the windows acknowledged, and D due to data using the expansion of the window by the slow-start exponential increase policy) sent in the current cycle. Assuming that there is bandwidth available to transmit this data upto the bottleneck, the worst case queue is $2D$. Observe that the total average load measured over an RTT is still twice the Link_bandwidth, however there is an extra instantaneous queue of $D = 1 \times \text{RTT} \times \text{Link\_bandwidth}$. This is because the entire queue builds up within a cell transmission time. In the case when TCP load is smooth, two cells are being input for every cell drained from the queue.

Observe that once the sources are rate-limited, the TCP burstiness due to ack compression is hidden from the ATM network because the excess cells are

buffered at the end-system, and not inside the ATM network. This is a significant performance benifit considering the fact that without rate-control the buffer requirement may be very large [119].

7. **Effect of VBR backgrounds:** The presence of higher priority background traffic implies that the ABR capacity is variable. There are two implications of the variation in capacity: **a)** the effect on the rate of TCP acks and the window growth, and, **b)** the effect on the switch rate allocation algorithm (ERICA).

   Part a): The effect of VBR background on the TCP "ack clock" (i.e., the rate of TCP acknowledgements (ACKs) is described below.

   - If VBR comes on during zero ABR load, it does not affect the TCP ACK clock because there are no ABR cells in the pipe. This is the case when VBR comes on during the idle periods of TCP, especially in the initial startup phase of TCP.
   - If VBR comes on during low load (initial load less than unity), *such that the resulting total load is less than unity,* it increases the TCP packet transmission time. Increased packet transmission time means that the inter-ACK time is increased (called *"ACK expansion"*). Increase in inter-ACK time slows down the window increase since the "ACK clock" runs slower now. Slower window increases imply reduced ABR load in the next cycle, irrespective of whether VBR is off or on.

     Specifically, if VBR goes off, the inter-ACK spacing starts decreasing by half every round trip (due to TCP window doubling every round trip). Since we have assumed that the system was in low load previously, this

change in VBR load does not affect the ABR load immediately. The TCP continues its exponential increase as if it started at the current window levels. The analysis is then similar to the case where VBR is absent. We are assuming that the source is not rate-limited by ABR during this phase. There is no excess ABR queues due to

- If VBR comes such that the resulting total load is greater than unity, then queues build up, and the ACK expansion occurs. The TCP load grows at a slower rate, and the ABR feedback mechanism throttles the source rates, leading to the excess TCP load being buffered at the sources. The excess network queue due to this case is $1 \times RTT \times VBRBandwidth$.

  TCP does experience variation in the rate of acknowledgements. The rate of acknowledgements determine how rapidly the TCP window grows. Note that if the sum of the windows is such that it fully loads the bottleneck (no idle periods), then the variable rate of acknowledgements only determines how quickly excess data is dumped onto the ATM sources by TCP. If the ABR sources are rate-limited, and have TCP queues built up, the excess data affects the queue at the source and not the network. The network queue is affected by the rate changes caused by the switch algorithm, as described in part b. If the source queue is close to zero, then the excess data dumped by TCP due to the variable "ack clock" affects the network queue as well. The effect of VBR on TCP ack clock is as follows:

  However, once the VBR load goes away, the ABR feedback (see part b) determines how many cells are admitted into the network and how many remain at the sources.

Part b): When ABR load goes away, then the switches see a sudden underload and allocate high rates to sources. Note that a switch which sees no cells in an interval or a small number of cells due to the fact that VBR load disappears is dealing with transient, incomplete metric information. As a result, if it overallocates rates, then sudden queue spikes are seen. In the worst case, the queues may grow unboundedly as a result of several such high priority VBR on/off phases. The buffering required under this condition is heavily dependent upon the switch algorithm. We present the problem, simulation results, modifications we made to the ERICA algorithm, and results with the improved algorithm in section 8.16 later in this chapter. We also model MPEG-2 traffic over VBR and study its effect on TCP traffic over ABR in section 8.17.

We have seen that the round trip time, feedback delay, the bandwidth variability caused by VBR, and the switch algorithm are key factors which determine the buffer requirements for TCP over ABR.

From items 4 (switch convergence time) and 5 (queue backlogs) above, we find that for file transfer over ABR, we require at least $3 \times RTT$ worth of buffer. Items 6 (two-way traffic) and 7 (VBR traffic) require a buffer of at least $5 \times RTT$. Note that the effect of the averaging interval parameter dominates in LANs (because it is much larger than RTT or feedback delay). Similarly, the effect of the feedback delay dominates in satellite networks because it can be much smaller than RTT. We substantiate our claims with simulation results, where we will observe that the buffer requirement is at most $3 \times RTT$.

Though the maximum ABR network queues are small, the queues at the sources are high. Specifically, the maximum sum of the queues in the source and the switches

is equal to the sum of the TCP window sizes of all TCP connections. In other words the buffering requirement for ABR becomes the same as that for UBR if we consider the source queues into consideration. This observation is true only in certain ABR networks. If the ATM ABR network is an end-to-end network, the source end systems can directly flow control the TCP sources. In such a case, the TCP will do a blocking send, i.e., and the data will go out of the TCP machine's local disk to the ABR source's buffers only when there is sufficient space in the buffers. The ABR service may also be offered at the backbone networks, i.e., between two routers. In these cases, the ABR source cannot directly flow control the TCP sources. The ABR flow control moves the queues from the network to the sources. If the queues overflow at the source, TCP throughput will degrade. We substantiate this in section 8.15.5

Note that we have studied the case of infinite traffic (like a large file transfer application) on top of TCP. In section 8.19, we show that bursty (idle/active) applications on TCP can potentially result in unbounded queues. However, in practice, a well-designed ABR system can scale well to support a large number of applications like bursty WWW sources running over TCP [115].

8.15 Factors Affecting Buffering Requirements of TCP over ATM-ABR Service

In this section we present sample simulation results to substantiate the preceding claims and analyses. We also analyze the effect of some important factors affecting the ABR buffering requirements. The key metric we observe is the maximum queue length. We look at the effect of VBR in two separate sections, the second of which deals with multiple MPEG-2 sources using a VBR connection.

All our simulations presented use the n Source configuration presented earlier. Recall that it has a single bottleneck link shared by n ABR sources. All links run at 155.52 Mbps and are of the same length. We experiment with the number of sources and the link lengths.

All traffic is unidirectional. A large (infinite) file transfer application runs on top of TCP for the TCP sources. N may assume values 1, 2, 5, 10, 15 and the link lengths 1000, 500, 200, 50 km. The maximum queue bounds also apply to configurations with heterogenous link lengths.

The TCP and ABR parameters are set in the same way as in the earlier simulations. For satellite round trip (550 ms) simulations, the window is set using the TCP window scaling option to 34000×2^8 bytes.

8.15.1 Effect of Number of Sources

In Table 8.2, we notice that although the buffering required increases as the number of sources is increased, the amount of increase slowly decreases. As later results will show, three RTTs worth of buffers are sufficient even for large number of sources. In fact, one RTT worth of buffering is sufficient for many cases: for example, the cases where the number of sources is small. The rate allocations among contending sources were found to be fair in all cases.

8.15.2 Effect of Round Trip Time (RTT)

From Table 8.3, we find that the maximum queue approaches

$3 \times$ RTT $\times$ link bandwidth,

particularly for metropolitan area networks (MANs) with RTTs in the range of 6 ms to 1.5 ms. This is because the RTT values are lower and in such cases, the effect

Number of Sources	RTT(ms)	Feedback delay(ms)	Max Q (cells)	Thoughput
5	30	10	10597 = 0.95*RTT	104.89
10	30	10	14460 = 1.31*RTT	105.84
15	30	10	15073 = 1.36*RTT	107.13

Table 8.2: Effect of number of sources

of switch parameters on the maximum queue increases. In particular, the ERICA averaging interval parameter is comparable to the feedback delay.

Number of Sources	RTT(ms)	Feedback Delay (ms)	Max Q size(cells)	Thoughput
15	30	10	15073 = 1.36*RTT	107.13
15	15	5	12008 = 2.18*RTT	108.00
15	6	2	6223 = 2.82*RTT	109.99
15	1.5	0.5	1596 = 2.89*RTT	110.56

Table 8.3: Effect of Round Trip Time (RTT)

8.15.3 LANs: Effect of Switch Parameters

In Table 8.4, the number of sources is kept fixed at 15. The averaging interval is specified as a pair (T, n), where the interval ends when either T ms have expired or N cells have been processed, whichever happens first. For the parameter values shown in the table, the number of cells determined the length of the averaging interval since under continuous traffic 1000 ATM cells take only 2.7 ms.

Averaging Interval (ms,cells)	RTT(ms)	Feedback Delay (ms)	Max Q size(cells)	Thoughput
(10,500)	1.5	0.5	2511	109.46
(10,1000)	1.5	0.5	2891	109.23
(10,500)	0.030	0.010	2253	109.34
(10,1000)	0.030	0.010	3597	109.81

Table 8.4: Effect of Switch Parameter (Averaging Interval)

From Table 8.4, we observe that the effect of the switch parameter, averaging interval, dominates in LAN configurations. The ERICA averaging interval is much greater than the RTT and feedback delay and it determines the congestion response time and hence the queue lengths. configurations. The ERICA averaging interval becomes much greater than

8.15.4 Effect of Feedback Delay

We conducted a 3×3 full factorial experimental design to understand the effect of RTT and feedback delays [66]. The results are summarized in Table 8.5. The thoughput figures for the last three rows (550 ms RTT) are not available since the throughput did not reach a steady state although the queues had stabilized.

Observe that the queues are small when the feedback delay is small and do not increase substantially with round-trip time. This is because the switch scheme limits the rate of the sources before they can overload for a substantial duration of time.

RTT(ms)	Feedback Delay (ms)	Max Q size(cells)	Thoughput
15	0.01	709	113.37
15	1	3193	112.87
15	10	17833	109.86
30	0.01	719	105.94
30	1	2928	106.9
30	10	15073	107.13
550	0.01	2059	NA
550	1	15307	NA
550	10	17309	NA

Table 8.5: Effect of Feedback Delay

8.15.5 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.

Source End System Queues in ABR

Table 8.6 shows the results with a 15-source configuration with link lengths of 1000 km (there is no VBR background). The link lengths yield a round trip time (propagation) of 30 ms and a feedback delay of 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 including the averaging feature for the number of sources and an averaging interval of (5 ms, 500 cells) as described in Section 8.16.1.

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 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 round trip time (indicated as "RTT" in the table). The round trip time for this configuration is 30 ms which corresponds to a "cell length" of 30 ms $\times$ 368 cells/ms = 11040 cells.

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

#	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

Table 8.6: Source Queues in ABR

In rows 1, 2 and 3 of Table 8.6, 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.

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 typically 3×RTT as discussed earlier in this chapter.

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 8.6 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 zero because the sources are rate-limited by the ABR mechanism. 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 [38]. 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.

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. 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 they have a VC per 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 the service providers. In general, UBR service requires more buffering in the switches than the ABR service.

8.15.6 Summary of buffering requirements for TCP over ABR

The main results of this section are:

1. A derivation for the buffer requirements of TCP over ABR is given. The factors which affect the buffer requirements are RTT, switch algorithm parameters, feedback delay, presence of VBR traffic, or two-way TCP traffic. For a switch algorithm like ERICA, the buffer requirements are about $3 \times RTT \times Link\_bandwidth$. The derivation is valid for infinite applications (like file transfer) running over TCP.

2. Once the ABR sources are rate-limited, the queues build up at the sources, and not inside the network. This has implications for ATM backbone networks where the edge routers either need large buffers, or need to implement some form of flow control with the TCP end system to avoid loss.

8.16 Effect of ON-OFF VBR Background Traffic

In this section we examine the effect of ON-OFF VBR background on the buffer requirements for TCP over ABR. We use the n source + VBR configuration as before. The parameter changes in the configuration are described below:

We use the ERICA+ scheme in the VBR simulations, and compare the performance with the ERICA scheme in certain cases. Recall that the ERICA+ scheme is an extension of ERICA which uses the queueing delay as a additional metric to calculate the feedback. 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). In certain cases, we use averaging schemes for the metrics used by ERICA, and a longer averaging interval: min(5 ms, 500 cells).

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.

8.16.1 Simulation Results

Table 8.7 shows the results of a 3x3 full-factorial experimental design [66] 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).

Effect of VBR ON-OFF Times

Rows 1,2 and 3 of Table 8.7 characterize large ON-OFF times (low frequency VBR). Observe that the (maximum) queues are small fractions of the round trip time. 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 8.7 characterize medium ON-OFF times. We observe that rows 5 and 6 have divergent (unbounded) queues. The effect of the ON-OFF time on the divergence is explained as follows. During the OFF time the switch experiences underload and may allocate high rates to sources. The duration of the OFF time

#	Duty Cycle(d) (ms)	Period (p) (cells)	Max Switch Q
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	DIVERGENT
6.	0.7	10	DIVERGENT
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 8.7: Effect of VBR ON-OFF Times

determines how long such high rate feedback is given to sources. In the worst case, the ABR load 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 8.7 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. ERICA+ however controls the queues to small values in these cases.

Effect of Feedback Delays with VBR

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 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 load due to the high rate feedback is seen at the switch, it takes at least one feedback delay to reduce the rates of the sources.

The experiments shown in Table 8.7 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 4,5 of Table 8.7), the queues may diverge.

Table 8.8 shows the effect of varying the feedback delay and round trip time. We select the divergent case (row 4) from Table 8.7 and vary the feedback delay and round trip time of the configuration.

Row 1 in Table 8.8 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

#	Feedback Delay(ms)	RTT (ms)	Duty Cycle (d)	Period (p) (ms)	Max Switch Q (cells)
1.	1 ms	3 ms	0.8	10 ms	4176 (0.4×RTT)
2.	5 ms	15 ms	0.8	10 ms	DIVERGES
3.	10 ms	30 ms	0.8	10 ms	DIVERGES

Table 8.8: Effect of Feedback Delay with VBR

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 8.8 have a feedback delay longer than the OFF time. This is one of the factors causing the divergence in the queues of these rows.

Effect of Switch Scheme with VBR

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 variation. The errors in the metrics are reflected in the feedback. The errors in the feedback result in queues. Switch schemes need to be robust to perform under such error-prone conditions. Another effect of errors is that the boundary conditions of the scheme are encountered often. The scheme must be designed to handle such conditions gracefully. We study the robustness issues in ERICA and make adjustments needed to reduce the effect of the variation.

As an example, consider the case when the VBR ON-OFF periods are very small (1 ms ON, 1 ms OFF). The resulting variation 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 without queue control cannot handle this level of variation.

The ERICA+ algorithm uses the queue length as a secondary metric to reduce the high allocation of rates. However, ERICA+ has a limit on how much it can reduce the allocation. Given sufficient variation, the limit can be reached. This means that even the minimum rate allocation by ERICA+ causes the queues to diverge. This reason, along with the discussion on ON-OFF times and feedback delays explains the divergent cases in Tables 8.7 and 8.8.

Reducing the Effects of Variation In ERICA+

We tackle these problems by reducing the effect of variation on the scheme measurements in three ways (described in detail in chapter 6):

1. First, we observe that one way to reduce variation in measurements is to measure quantities over longer intervals. Longer intervals yield averages which have less variance. However, making the intervals too long increases the response time, and queues may build up in the interim.

2. Second, we average the measurements over several successive intervals. 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_a). We re-examine how the scheme depends on these metrics and design an appropriate averaging technique for each of them.

    - The overload factor (z) is used 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. If the overload factor (z) is underestimated, the VC share increases. The overload factor is usually not overestimated. However, if the interval length is small, the estimated values may have high variation.

The overload factor (z) can suddenly change in an interval if the load or capacity in that interval changes due to the variation. The out-of-phase effect of TCP may lead to no cells being seen in the forward direction ($z = 0$, a huge underestimate !), while BRM cells are seen in the reverse direction. The switch will then allocate a high rate in the feedback it writes to the BRM cell.

We have designed two averaging schemes for the overload as described in chapter 6. Both schemes use an averaging parameter "α_z". The first scheme is similar to the technique of exponential averaging technique for a random variable. However, it differs in that it resets the averaging mechanism whenever the instantaneous value of overload is measured to be zero or infinity. The second scheme does not ignore the outlier values (zero or infinity) of the overload factor. Further, it averages the overload by separately averaging the input rate and capacity, and then taking the ratio of the averages. It can be shown [66] that this is theoretically the right way to average a ratio quantity like overload.

- The number of active sources (N_a) is used to calculate a minimum fairshare that is given to any source. If N_a is underestimated, then the minimum fairshare is high leading to overallocation. If N_a is overestimated, then the

minimum fairshare is low. This may result in slower transient response, but does not result in overallocation.

The number of active sources can fluctuate if some sources are not seen in an interval. Further, due to the clustering effect of TCP, cells from just a few VCs may be seen in an interval leading to an underestimate of N_a.

In averaging N_a, the scheme maintains an activity level for each source. The activity level of the source is set to one when any cell of the source is seen in the interval. However, when no cell from a source is seen in an interval, the scheme "decays" the activity level of the source by a factor, "α_n" (also called $DecayFactor$). Hence, the source becomes inactive only after many intervals. A recommended value of α_n is 0.9. Roughly, the N_a measured with this value of α_n is approximately equal to the N_a measured without averaging over an averaging interval 8 or 9 times larger than the current averaging interval.

3. Third, we modify the response to boundary conditions of the scheme. This allows the scheme to handle the boundary conditions gracefully. Specifically, the number of active sources is set to one if it is measured to be below one. The second method of overload factor averaging does not allow the overload factor be zero or infinity. However, outlier measurements are not ignored in the averaging method.

The ERICA+ scheme with these modifications controls the ABR queues without overly compromising on TCP throughput. Table 8.9 shows the results of representative experiments using these features.

#	Averaging Interval (T ms, n cells)	Averaging of Na on ? ($\alpha_n = 0.9$)	Averaging of z on ? ($\alpha_z = 0.2$)	d	p(ms)	Max Switch Queue (cells)
1.	(1,100)	YES	YES	0.7	20	5223
2.	(5,500)	YES	NO	0.7	20	5637

Table 8.9: Effect of Switch Scheme

Row 1 shows the performance with the averaging of Na and z turned on on a formerly divergent case. 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. Hence, we try the value (5 ms, 500 cells) 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.

8.16.2 Summary of ON-OFF VBR background effects

In this section, we have studied the impact of ON-OFF VBR background traffic on switch buffering for ABR service carrying TCP traffic. We find that the ON-OFF times, the feedback delays, and a switch scheme sensitive to variation in ABR load and capacity may combine to create worst case conditions where the ABR queues diverge. We have motivated three enhancements to the ERICA+ scheme. The modifications reduce the effect of the variation and allow the convergence of the ABR queues, without compromising on the efficiency. In the next section, we shall examine the effect of VBR carrying long-range dependent traffic (similar to multiplexed MPEG-2 traffic) on ABR, and show that the buffer requirements are unchanged.

8.17 Effect of Long-Range Dependent (LRD) VBR background traffic

We have studied the ABR model extensively with different source traffic patterns like persistent sources, ON-OFF bursty sources, ping pong sources, TCP sources and source-bottlenecked VCs. Many of these studies have also considered the performance in the presence of ON-OFF VBR background traffic.

In reality, VBR consists of multiplexed compressed audio and video application traffic, each shaped by leaky buckets at their respective Sustained Cell Rate (SCR) and Peak Cell Rate (PCR) parameters. Compressed video has been shown to be long-range dependent by nature [13]. Compressed audio and video streams belonging to a single program are expected to be carried over an ATM network using the MPEG-2 Transport Stream facility as outlined in reference [37].

In this section, we first present a model of multiplexed MPEG-2 transport streams carried over ATM using the VBR service. Each stream exhibits long-range dependence, i.e., correlation over large time scales. We then study the effect of this VBR background on ABR connections carrying TCP file transfer applications on WAN and satellite configurations. The effect of such VBR traffic is that the ABR capacity is highly variant. We find that a proper switch algorithm like ERICA+ can tolerate this variation in ABR capacity while maintaining high throughput and low delay. We will present simulation results for terrestrial and satellite configurations.

8.17.1 Overview of MPEG-2 over ATM

In this section, we give a short introduction to the MPEG-2 over ATM model and introduce some MPEG-2 terminology. For a detailed discussion, see reference [111].

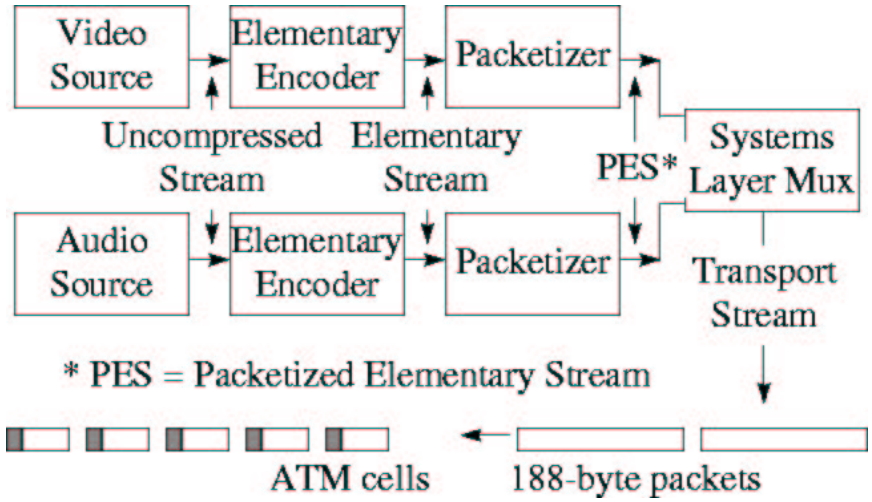


Figure 8.13: Overview of MPEG-2 Transport Streams

The MPEG-2 standard specifies two kinds of streams to carry coded video: the "Transport Stream" and the "Program Stream". The latter is used for compatibility with MPEG-1 (used for stored compressed video/audio), while the former is used to carry compressed video over networks which may not provide an end-to-end constant delay and jitter-free abstraction.

A Transport Stream can carry several programs multiplexed into one stream. Each program may consist of several "elementary streams," each containing MPEG-2 compressed video, audio, and other streams like close-captioned text, etc.

Figure 8.13 shows one such program stream formed by multiplexing a compressed video and a compressed audio elementary stream. Specifically, the figure shows the uncompressed video/audio stream going through the MPEG-2 elementary encoder to form the elementary stream. Typically, the uncompressed stream consists of frames generated at constant intervals (called "frame display times") of 33 ms (NTSC format) or 40 ms (PAL format). These frames (or "Group of Pictures" in MPEG-2

terminology) are called "Presentation Units." MPEG-2 compression produces three different types of frames: I, P and B frames, called "Access Units," as illustrated in Figure 8.14.

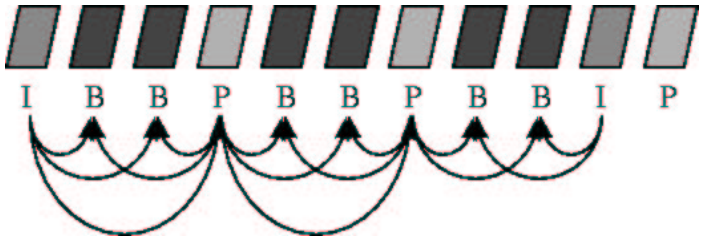


Figure 8.14: The I, P and B frames of MPEG-2

I (Intra-) frames are large. They contain the base picture, autonomously coded without need of a reference to another picture. They might take about 4-5 frame display times (approximately 160 ms) to be transmitted on the network depending upon the available rate.

P (Predictive-) frames are medium-sized. They are coded with respect to previous I or P frame. Transmission times for P frames is typically about 0.5-1 frame display times.

B (Birectionally predicted-) frames are very small. They are coded with respect to previous and later I or P frames and achieve maximum compression ratios (200:1). Transmission times for B frames is typically about 0.2 frame display times or even less.

As shown in Figure 8.13, the access units are packetized to form the "Packetized Elementary Stream (PES)". PES packets may be variable in length. The packetization process is implementation specific. PES packets may carry timestamps (called

Presentation Timestamps (PTS) and Decoding Timestamps (DTS)) for long-term synchronization. The MPEG-2 standard specifies that PTS timestamps must appear at least once every 700 ms.

The next stage is the MPEG-2 Systems Layer which does the following four functions. First, it creates fixed size (188 byte) transport packets from PES packets. Second, the transport packets of different PESs belonging to one program are identified as such in the transport packet format. Third, it multiplexes several such programs to create a single Transport Stream. Fourth, it samples a system clock (running at 27 MHz) and encodes timestamps called "MPEG2 Program Clock References" (MPCRs, see [37]) in every multiplexed program. The time base for different programs may be different.

The MPCRs are used by the destination decoder to construct a Phase Locked Loop (PLL) and synchronize with the clock in the incoming stream. The MPEG-2 standard specifies that MPCRs must be generated at least once every 100 ms. Due to AAL5 packetization considerations, vendors usually also fix a maximum rate of generation of MPCRs to 50 per second (i.e. no less than one MPCR per 20 ms).

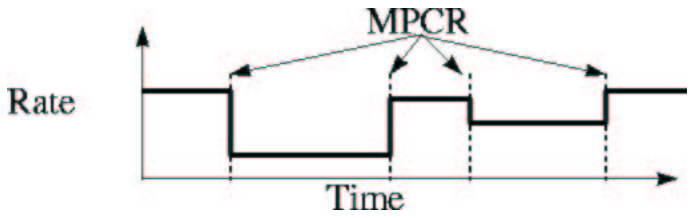


Figure 8.15: Piecewise constant nature of MPEG-2 Single Program Transport Streams

The key point is that the MPEG-2 rate is piecewise-CBR. As shown in Figure 8.15, the program's rate (not the transport stream's rate) is constant between successive MPCRs. The maximum rate is bounded by a peak value (typically 15 Mbps for HDTV quality compressed video [111]). The choice of the rates between MPCRs is implementation specific, but in general depends upon the buffer occupancy, and the rate of generation of the elementary streams.

The transport stream packets are encapsulated in AAL5 PDU with two transport stream packets in a single AAL5 PDU (for efficiency). The encapsulation method does not look for MPCRs in a transport packet and might introduce some jitter in the process. Alternate methods and enhancements to the above method have been proposed [111, 42].

An ATM VBR connection can multiplex several transport streams, each containing several programs, which in turn can contain several elementary streams. We model the multiplexing of several transport streams over VBR. But in our model, we will have only one program per Transport Stream (called the "Single Program Transport Stream" or "SPTS").

MPEG-2 uses a constant end-to-end delay model. The decoder at the destination can use techniques like having a de-jittering buffer, or restamping the MPCRs to compensate for network jitter, [111]. There is a Phase Locked Loop (PLL) at the destination which locks onto the MPCR clock in the incoming stream. The piecewise-CBR requirement allows the recovered clock to be reliable. Engineering of ATM VBR VCs to provide best service for MPEG-2 transport streams and negotiation of rates (PCR, SCR) is currently an important open question.

8.17.2 VBR Video modeling

There have been several attempts to model compressed video, see references [13, 36, 23] and references therein. Beran et al [13] show that long-range dependence is an inherent characteristic of compressed VBR video. But, they do not consider MPEG-2 data. Garrett and Willinger [36] show that a combination of distributions is needed to model VBR video. Heyman and Lakshman [23] argue that simple markov chain models are sufficient for traffic engineering purposes even though the frame size distribution may exhibit long-range dependence.

The video traffic on the network may be affected further by the multiplexing, renegotiation schemes, feedback schemes and the service category used. Examples of renegotiation, feedback schemes and best-effort video delivery are found in the literature, [41, 87, 25].

We believe that a general model of video traffic on the ATM network is yet to be discovered. In this paper, we are interested in the performance of ABR carrying TCP connections when affected by a long-range dependent, highly variable VBR background. We hence need a model for the video background. We have attempted to design the model to resemble the MPEG-2 Transport Stream.

There are three parameters in the model: the compressed video frame size, the inter-MPCR interval lengths, and the rates in these inter-MPCR intervals. In our model, the inter-MPCR intervals are uniformly distributed and the rates in the inter-MPCR intervals are long-range dependent. In real products, the rates are chosen depending upon the buffer occupancy at the encoder, which in turn depends upon the frame sizes of the latest set of frames generated. Further, the range of inter-MPCR intervals we generate follows implementation standards. We believe that this models

the MPEG-2 Transport Stream, and still incorporates the long-range dependence property in the video streams. The effect of this VBR model on ABR is to introduce high variation in ABR capacity. As we shall see, the ERICA+ algorithm deals with the variation in ABR capacity and successfully bounds the maximum ABR queues, while maintaining high link utilization.

8.17.3 Modeling MPEG-2 Transport Streams over VBR

We model a "video source" as consisting of a transport stream generator, also called encoder (E) and a network element (NE). The encoder produces a Transport Stream as shown in Figure 8.13 and discussed in section 8.17.1 . In our model, the Transport Stream consists of a single program stream. The network element encapsulates the transport packets into AAL5 PDUs and then fragments them into cells. The output of the network element (NE) goes to a leaky bucket which restricts the peak rate to 15 Mbps. This leaky bucket function can alternatively be done in the encoder, E (which does not send transport packets beyond a peak rate).

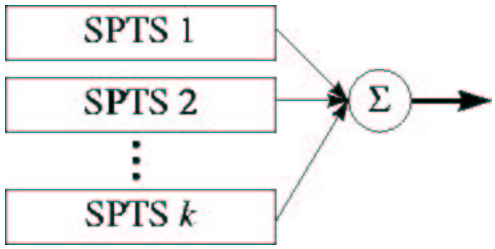


Figure 8.16: Multiplexing MPEG-2 Single Program Transport Streams (SPTSs) over VBR

Several (N) such video sources are multiplexed to form the VBR traffic going into the network as shown in Figure 8.16. Each encoder generates MPCRs uniformly

distributed between 20 ms and 100 ms. The reason for this choice (of maximum and minimum MPCRs) is explained in section 2. The rate of an encoder is piecewise-constant between successive pairs of MPCRs.

We generate the rates as follows. We choose the rate such that the sequence of rate values is long-range dependent. Specifically, we use a fast-fourier transform method [96] to generate the fractional gaussian noise (FGN) sequence (an independent sequence for each source). We ignore values above the maximum rate to 15 Mbps and below the minimum rate (0 Mbps). This reason for this choice is discussed in the following section. We choose different values of mean and standard deviation for the generation procedure. When we generate an inter-MPCR interval Ti and a corresponding rate Ri, the video source sends cells at a rate Ri uniformly spaced in the interval Ti. Due to the ignoring of some rate values, the actual mean of the generated stream may be slightly greater or lesser than the input means. We later measure the actual mean rate and use it to calculate the efficiency metric.

Though each video source sends piecewise-CBR cell streams, the aggregate VBR rate need not be piecewise-CBR. It has a mean (SCR) which is the sum of all the individual means. Similarly, it has a maximum rate (PCR) which is close to the sum of the peak rates (15 Mbps) of the individual video streams. These quantities depend upon the number of video sources. In our model, we use N equal to 9 to ensure that the PCR is about 80% of total capacity. VBR is given priority at any link, i.e, if there is a VBR cell, it is scheduled for output on the link before any waiting ABR cells are scheduled. Further, since each video stream is long-range dependent, the composite VBR stream is also long-range dependent. Therefore, the composite VBR stream and the ABR capacity has high variation.

8.17.4 Observations on the Long-Range Dependent Traffic Generation Technique

The long-range dependent generation technique described in [96] can result in negative values and values greater than the maximum possible rate value. This occurs especially when the variation of the distribution is high (of the order of the mean itself). Fortunately, there are a few approaches in avoiding negative values and bounding values within a maximum in such sequences. We considered these approaches carefully before making a choice.

The first approach is to generate a long-range dependent sequence $x_1, x2, ..., xn$ and then use the sequence $e^{x_1}, e^{x_2}, ..., e^{x_n}$ in our simulation. The values e^{x_i} is rounded off to the nearest integer. This method always gives zero or positive numbers. The new distribution still exhibits long-range dependence, though it is no longer a fractional gaussian noise (FGN) (like the originally generated sequence) [96]. Another problem is that all significant negative values are truncated to zero leading to an impulse at zero in the new probability density function (pdf). Further, the mean of the new sequence is not the exponentiated value of the old mean. This makes it difficult to obtain a sequence having a desired mean.

A second technique is to avoid exponentiation, but simply truncate negative numbers to zero. This approach again has the problem of the pdf impulse at zero. Also the mean of the entire distribution has increased.

The third technique is a variation of the second, which truncates the negative numbers to zero, but subtracts a negative value from the subsequent positive value.

This approach is aimed to keep the mean constant. But, it not only has the side-effect of inducing a pdf impulse at zero, but also changes the shape of the pdf, thus increasing the probability of small positive values.

The fourth and final technique is to simply ignore negative values and values greater than the maximum. This approach keeps the shape of the positive part of the pdf intact while not introducing a pdf impulse at zero. If the number of negative values is small, the mean and variance of the distribution would not have changed appreciably. Further, it can be shown that the new distribution is still long-range dependent.

We choose the fourth approach (of ignoring negative values and values greater than the maximum) in our simulations.

8.18 Simulation Configuration and Parameters

We use the n Source + VBR configuration described in section 8.7 earlier in this chapter. Recall that the configuration has a single bottleneck link shared by the N ABR sources and a VBR VC carrying the multiplexed stream. Each ABR source is a large (infinite) file transfer application using TCP. All traffic is unidirectional. All links run at 149.76 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 are 1000 km in WAN simulations. In satellite simulations, the feedback delay may be 550 ms (corresponds to a bottleneck after the satellite link) or 10 ms (corresponds to a bottleneck before the satellite link). This is illustrated in Figures 8.17 and 8.18.

For the video sources, we choose means and standard deviations of video sources to have three sets of values (7.5 Mbps, 7 Mbps), (10 Mbps, 5 Mbps) and (5 Mbps, 5 Mbps). This choice ensures that the variance in all cases is high, but the mean varies and hence the total VBR load varies. The number of video sources (N) is 9 which means that the maximum VBR load is 80% of 149.76 Mbps link capacity. As discussed later the effective mean and variance (after bounding the generated value to within 0 and 15 Mbps) may be slightly different and it affects the efficiency measure. We also compare certain results with those obtained using an ON-OFF VBR model described in section 8.16.

The Hurst parameter which determines the degree of long-range dependence for each video stream is chosen as 0.8 [13].

Recall that when TCP data is encapsulated over ATM, 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. The maximum throughput of TCP over raw ATM is (512 bytes/(12 cells $\times$ 53 bytes/cell)) = 80.5%. Further in ABR, we send FRM cells once every Nrm (32) cells. Hence, the maximum throughput is $31/32 \times 0.805 = 78\%$ of ABR capacity. For example, when the ABR capacity is 149.76 Mbps, the maximum TCP payload rate is 116.3 Mbps. Similarly, for a MSS of 9140 bytes, the maximum throughput is 87% of ABR capacity.

We use a metric called "efficiency" which is defined as the ratio of the TCP throughput achieved to the maximum throughput possible. As defined above the

maximum throughput possible is 0.78×(mean ABR capacity). The efficiency is calculated as follows. We first measure the the aggregate mean VBR rate (since it is not the sum of the individual mean rates due to bounding the values to 0 and 15 Mbps). Subtract it from 149.76 Mbps to get the mean ABR capacity. Then multiply the ABR capacity by 0.78 (or 0.87) to get the maximum possible throughput. We then take the ratio of the measured TCP throughput and this calculated value to give the efficiency.

8.18.1 Effect of High Variance and Total VBR Load

In this section, we present simulation results where we vary the mean and the standard deviation of the individual video sources such that the total variance is always high, and the total maximum VBR load varies.

In Table 8.10, and Table 8.11, we show the maximum queue length, the total TCP throughput, VBR throughput, ABR throughput, and efficiency for three combinations of the mean and standard deviation. Table 8.11 is for TCP MSS = 512 bytes, while Table 8.11 is for TCP MSS = 9140 bytes.

	Video Sources		ABR Metrics		
#	Mean per-source rate (Mbps)	Standard Deviation (Mbps)	Max Switch Q (cells)	Total TCP T'put	Efficiency (% of Max throughput)
1.	5	5	6775 (1.8×F/b Delay)	68.72 Mbps	94.4%
2.	7.5	7	7078 (1.9×F/b Delay)	59.62 Mbps	94.1%
3.	10	5	5526 (1.5×F/b Delay)	82.88 Mbps	88.4%

Table 8.10: Effect of Variance and VBR Load: MSS = 512 bytes

	Video Sources		ABR Metrics		
#	Mean per-source rate (Mbps)	Standard Deviation (Mbps)	Max Switch Q (cells)	Total TCP Throughput	Efficiency (% of Max throughput)
1.	5	5	5572 (1.5×F/b Delay)	77.62 Mbps	95.6%
2.	7.5	7	5512 (1.5×F/b Delay)	67.14 Mbps	95%
3.	10	5	5545 (1.5×F/b Delay)	56.15 Mbps	95.6%

Table 8.11: Effect of Variance and VBR Load: MSS = 512 bytes

Observe that the measured mean VBR thoughput (column 6) is the same in corresponding rows of both the tables. This is because irrespective of ABR load, VBR load is given priority and cleared out first. Further, by bounding the MPEG-2 SPTS source rate values between 0 and 15 Mbps, we ensure that the total VBR load is about 80of the link capacity.

For row 1, measured VBR throughput (column 6) was 56.44 Mbps (against $9 \times 5 = 45$ Mbps expected without bounding). For row 2, it was 68.51 Mbps (against $9 \times 7.5 = 67.5$ Mbps expected without bounding). For row 3, it was 82.28 Mbps(against $9 \times 10 = 90$ Mbps expected without bounding). Observe that when the input mean is higher, the expected aggregate value is lower and vice-versa.

The efficiency values are calculated using these values of total VBR capacity. For example, in row 1 of Table 8.10, the ABR throughput is is 149.76 - 56.44 = 93.32 Mbps. For a MSS of 512, the maximum TCP thoughput is 78% of ABR throughput = 72.78 Mbps (not shown in the table). Given that TCP thoughput achieved is 68.72 Mbps (Column 5), the efficiency is 68.72/72.78 = 94.4%. For Table 8.11, since the

MSS is 9140 bytes, the maximum TCP thoughput is 87% of ABR throughput, and this is the value used to compare the total TCP throughput against.

Observe that the efficiency achieved in all cases is high (above 90%) in spite of the high variation in ABR capacity. Also observe that the total TCP throughput is higher (as well as the efficiency) for TCP MSS = 9140 bytes in all cases.

The maximum queue length is controlled to about three times the feedback delay (or one round trip time) worth of queue. The feedback delay for this configuration is 10 ms, which corresponds to (10 ms) $\times$ (367 cells/ms) = 3670 cells worth of queue when the network is on the average overloaded by a factor of 2 (as is the case with TCP). The round-trip time for this configuration is 30 ms.

The queue length is higher when the mean per-source rate is lower (i.e., when the average ABR rate is higher). This is explained as follows. Whenever there is variation in capacity, the switch algorithm may make errors in estimating the average capacity and may overallocate rates temporarily. When the average ABR capacity is higher, each error in allocating rates will result in a larger backlog of cells to be cleared than for the corresponding case when the average ABR capacity is low. The combination of these backlogs may result in a larger maximum queue before the long-term queue reduction mechanism of the switch algorithm reduces the queues.

8.18.2 Comparison with ON-OFF VBR Results

Recall that in section 8.16 we have studied the behavior of TCP over ABR in the presence of ON-OFF VBR sources. We studied ranges of ON-OFF periods from 1 ms through 100 ms. Further, we had looked at results where the ON period was not equal to the OFF period. The worst cases were seen in the latter simulations.

However, with modifications to ERICA+ and a larger averaging interval we found that the maximum switch queue length was 5637 cells. This experiment has a duty cycle of 0.7 and a period of 20ms i.e., the ON time was 14 ms and the off time was 6 ms. Since we use the same switch algorithm parameters in this study, we can perform a comparison of the two studies.

We observe that, even after the introduction of the long-range dependent VBR model, the queues do not increase substantially (beyond one round trip worth of queues) and the efficiency remains high (around 90%). This is because the ERICA+ switch algorithm has been refined and tuned to handle variation in the ABR capacity and ABR demand. These refinements allow the convergence of the ABR queues, without compromising on the efficiency.

8.18.3 Satellite simulations with Short Feedback Delay

In this section and the next, we repeat the experiments with some links being satellite links. In the first set of simulations, we replace the bottleneck link shared by 15 sources with a satellite link as shown in Figure 8.17. The links from the second switch to the destination nodes are 1 km each. The total round trip time is 550 ms, but the feedback delay remains 10 ms.

Table 8.12 and Table 8.13 (similar to Tables 8.10 and 8.11) show the maximum switch queue length, the total TCP throughput, VBR throughput, ABR throughput, and efficiency for three combinations of the mean and standard deviation. Table 8.12 is for TCP MSS = 512 bytes, while Table 8.13 is for TCP MSS = 9140 bytes.

Note that the TCP startup time in this configuration is large because the round trip time (550 ms) is large and TCP requires multiple round trips to be able to use its

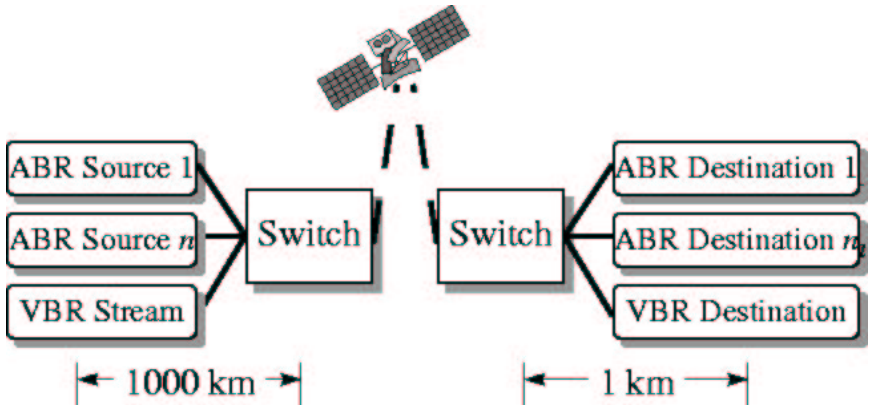


Figure 8.17: The "N Source + VBR" Configuration with a satellite link

full capacity. However, the effect on total TCP throughput is minimal since there is no loss and the feedback delays are small (10 ms) compared to round trip time, allowing ABR to control sources more effectively. Throughputs are high, and efficiency values are high.

	Video Sources		ABR Metrics				
#	Avg Src rate (Mbps)	STD (Mbps)	Max Switch Q (cells)	Total TCP T'put	VBR T'put	ABR T'put	Effcy (% Max T'put)
1.	5	5	5537 (1.5×f/b)	74.62	47.33	102.43	93.4%
2.	7.5	7	4157 (1.1×f/b)	67.34	57.23	92.53	93.3%
3.	10	5	3951 (1.1×f/b)	60.08	67.55	82.21	93.7%

Table 8.12: Maximum Queues for Satellite Networks with Short Feedback Delay: MSS=512 bytes

	Video Sources		ABR Metrics				
#	Avg Src rate (Mbps)	STD (Mbps)	Max Switch Q (cells)	Total TCP T'put	VBR T'put	ABR T'put	Effcy (% Max T'put)
1.	5	5	11294 (3×f/b delay)	84.72	47.33	102.43	95.1%
2.	7.5	7	9074 (2.5×f/b delay)	76.49	57.23	92.53	95.0%
3.	10	5	6864 (1.9×f/b delay)	68.18	67.55	82.21	95.3%

Table 8.13: Maximum Queues for Satellite Networks with Short Feedback Delay : MSS=9140 bytes

The tables shows that maximum queues are small (in the order of three times the feedback delay), irrespective of the mean and variance. In such satellite configurations, we observe that the feedback delay is the dominant factor (over round trip time) in determining the maximum queue length. As discussed earlier, one feedback delay of 10 ms corresponds to 3670 cells of queue for TCP.

8.18.4 Satellite simulations with Long Feedback Delay

In our second set of satellite simulations, we examine the effect of longer feedback delays. Consider a switch A at the end of a satellite link or a switch downstream of A. It will have a feedback delay of about 550 ms. This is the scenario we model. We form a new configuration as shown in Figure 8.18 by replacing the links in the feedback path to sources with satellite link. All other links are of length 1 km each. As a result, the round trip time and the feedback delay are both approximately equal to 550 ms.

Tables 8.14 and 8.15 (similar to Tables 8.10 and 8.11) show the maximum switch queue length, the total TCP throughput, VBR throughput, ABR throughput, and

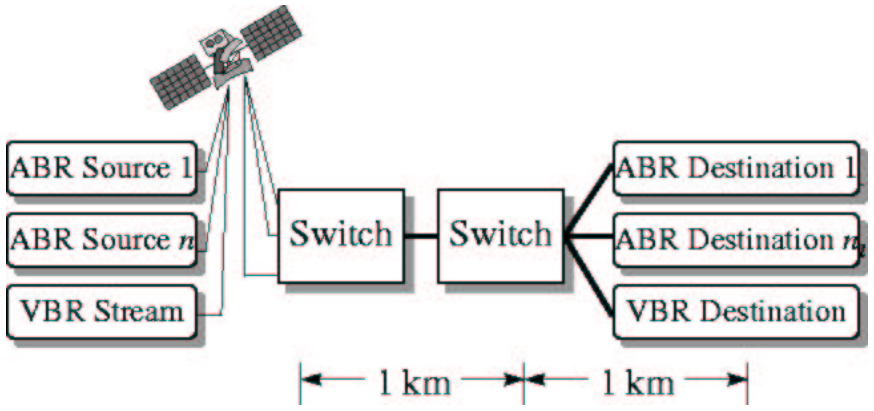


Figure 8.18: The "N Source + VBR" Configuration with satellite links and long feedback delays

efficiency for three combinations of the mean and standard deviation. Table 8.14 is for TCP MSS = 512 bytes, while Table 8.15 is for TCP MSS = 9140 bytes.

Observe that the queue lengths are quite large, while the total TCP throughput and efficiency are smaller (by 10-15%) compared to the values in Tables 8.12 and 8.13 (1000 km feedback delay cases) respectively. The total queue is still a small multiple of the feedback delay or RTT (a feedback delay of 550 ms corresponds to 201850 cells). This indicates that satellite switches need to provide at least so much buffering to avoid loss on these high delay paths. A point to consider is that these large queues should not be seen in downstream workgroup or WAN switches, because they will not provide so much buffering. Satellite switches can isolate downstream switches from such large queues by implementing the VSVD option as described in chapter 10.

	Video Sources		ABR Metrics				
#	Avg Src rate (Mbps)	STD (Mbps)	Max Switch Q (cells)	Total TCP T'put	VBR T'put	ABR T'put	Effcy (% Max T'put)
1.	5	5	255034 (1.3×f/b delay)	63.4	47.33	102.43	79.35%
2.	7.5	7	189276 (0.9×f/b delay)	59.8	57.23	92.53	82.86%
3.	10	5	148107 (0.7×f/b delay)	53.4	67.55	82.21	83.33%

Table 8.14: Maximum Queues for Satellite Networks with Long Feedback Delay: MSS=512 bytes

	Video Sources		ABR Metrics				
#	Avg Src rate (Mbps)	STD (Mbps)	Max Switch Q (cells)	Total TCP T'put	VBR T'put	ABR T'put	Effcy (% Max T'put)
1.	5	5	176007 (0.9×f/b delay)	71.92	47.33	102.43	80.70%
2.	7.5	7	252043 (1.3×f/b delay)	67.86	57.23	92.53	84.29%
3.	10	5	148646 (0.7×f/b delay)	59.33	67.55	82.21	82.95%

Table 8.15: Maximum Queues for Satellite Networks with Long Feedback Delay: MSS=9140 bytes

8.18.5 Summary of the effect of long-range dependent VBR

In this section, we have described how to model several multiplexed MPEG-2 video sources over VBR. Compressed video sources exhibit long-range dependence in the traffic patterns they generate. The effect of this long-range dependence is to introduce high variation in the ABR capacity. However a good switch scheme like ERICA+ is sufficient to handle this variation in ABR capacity. This results in

controlled ABR queues and high utilization. The maximum ABR queue length is a function of the feedback delay and round trip time. This implies that switches terminating satellite links should provide buffers proportional to the length of the satellite link in order to deliver high performance. Further, if they implement the VSVD option, they can isolate downstream workgroup switches from the effects of the long delay satellite path. We also briefly survey VBR video modeling techniques, the MPEG-2 over ATM approach, and propose a model for MPEG-2 video over VBR which incorporates the long-range dependence property in compressed video.

8.19 Effect of bursty TCP applications

In a related work [115], we have studied the effect of bursty applications running on top of TCP. An example of such an application is the World Wide Web application. The WWW application sets up TCP connections for its data transfers [33]. The WWW application differs from a large file transfer application in that while the latter looks like an "infinite or persistant" application to TCP, the former looks like a "bursty" application (with active and idle transmission periods). The effect of this on traffic management is described below.

TCP increases its "congestion window" as it receives acknowledgements for segments correctly received by the destination. If the application (eg. file transfer or WWW server/client) has data to send, it transmits the data. Otherwise, the window remains open until either the application has data to send or TCP times out (using a timer set by its RTT estimation algorithm). If the timer goes off, TCP reduces the congestion window to one segment (the minimum possible), and rises exponentially ("slow start") once the source becomes active again.

On the other hand, if the application remains idle for a period smaller than the timeout, the window is still open when the source becomes active again. If acknowledgements (corresponding to the data sent) are received within this idle interval, the window size increases further. Since no new data is sent during the idle interval, the usable window size is larger. The effect is felt when the application sends data in the new burst. Such behavior is possible by WWW applications using the HTTP/1.1 standard [33].

When TCP carrying such a WWW application runs over ATM, the burst of data is simply transferred to the network interface card (NIC). Assuming that each TCP connection is carried over a separate ABR VC, the data burst is sent into the ATM network at the VC's ACR. Since this VC has been idle for a period shorter than the TCP timeout (typically 500 ms for ATM LANs and WANs), it is an "ACR retaining" VC. Source End System (SES) Rule 5 specifies that the ACR of such a VC be reduced to ICR if the idle period is greater than parameter ADTF (which defaults to 500 ms). With this default value of ADTF, and the behavior of the TCP application, we are in a situation where the ACR is not reduced to ICR. This situation can be potentially harmful to the switches if ACRs are high and sources simultaneously send data after their idle periods.

Observe that an infinite application using TCP over ABR does not send data in such sudden bursts. As discussed in previous sections, the aggregate TCP load at most doubles every round trip time (since two packets are inserted into the network for every packet transmitted, in the worst case). Bursty TCP applications may cause the aggregate ABR load to more than double in a round trip time.

Note that the service capabilities in such a situation is also affected by a Use-it-or-Lose-it (UILI) implementation at the source or switch (as described in chapter 7). The UILI mechanism would reduce the source rate of the VC's carrying bursty TCP connections, and hence control the queues at the network switches. The effect of UILI in such conditions is for future study.

However, it has been shown that such worst case scenarios may not appear in practice due to the nature of WWW applications and the ABR closed-loop feedback mechanism [115]. Note that since the WWW traffic exhibits higher variation, techniques like averaging of load, and compensation for residual error (queues) as described in section 8.16.1 need to be used to minimize the effects of load variation. In summary, though bursty applications on TCP can potentially result in unbounded queues, a well-designed ABR system can scale well to support a large number of applications like bursty WWW sources running over TCP.

8.20 Summary of TCP over ABR results

This section unifies the conclusions drawn in each of the sections in this chapter (see sections 8.13, 8.15.6, 8.16.2, 8.18.5, and 8.19). In brief, the ABR service is an attractive option to support TCP traffic scalably. It offers high throughput for bulk file transfer applications and low latency to WWW applications. Further, it is fair to connections, which means that access will not be denied, and performance will not be unnecessarily degraded for any of the competing connections. This chapter shows that it is possible to achieve zero cell loss for a large number of TCP connections with a small amount of buffers. Hence, the ABR implementators can tradeoff the complexity

of managing buffers, queueing, scheduling and drop policies with the complexity of implementing a good ABR feedback scheme.

In section 8.13 we first noted that TCP can achieve maximum throughput over ATM if switches provide enough buffering to avoid loss. The study of TCP dynamics over the ABR service showed that initially when the network is underloaded or the TCP sources are starting to send new data, they are limited by their congestion window sizes (window-limited), rather than by the network-assigned rate (rate-limited). When cell losses occur, TCP loses throughput due to two reasons - **a)** a single cell loss results in a whole TCP packet to be lost, and, **b)** TCP loses time due to its large timer granularity. Intelligent drop policies (like avoiding drop of "End of Message (EOM)" cells, and "Early Packet Discard (EPD)" can help improve throughput). A large number of TCP sources can increase the total thoughput because each window size is small and the effect of timeout and the slow start procedure is reduced. We also saw that the ATM layer "Cell Loss Ratio (CLR)" metric is not a good indicator of TCP throughput loss. Further, we saw that the switch buffers should not be dimensioned based on the ABR Source parameter "Transient Buffer Exposure (TBE)". Buffer dimensioning should be based upon the performance of the switch algorithm (for ABR), and the round trip time.

In section 8.15.6, we summarized the derivation and simulation of switch buffer requirements for maximum throughput of TCP over ABR. The factors affecting the buffer requirements are round trip time, switch algorithm parameters, feedback delay, presence of VBR traffic, or two-way TCP traffic. For a switch algorithm like ERICA, the buffer requirements are about $3 \times RTT \times Link\_bandwidth$. The derivation is valid for infinite applications (like file transfer) running over TCP. Though the queueing

inside the ATM network can be controlled with the ABR service, in a heterogeneous network environment, the cells belonging to TCP streams may queue at the edge routers, i.e., at the entrance to the ATM network. Some form of end-to-end flow control involving the TCP end system is still necessary to avoid cell loss under such conditions.

In section 8.16.2 and 8.18.5, we studied the effect of the VBR background traffic patterns on the buffer requirements for TCP over ABR. The effect of the background traffic is to create variation in ABR capacity. The switch algorithm needs to be robust to handle the variation in ABR capacity (due to VBR) and in ABR demand (due to TCP dynamics). We motivate three enhancements to the ERICA+ scheme which reduce the effect of the variation and allow the convergence of the ABR queues, without compromising on efficiency. We then use a model of several multiplexed MPEG-2 video sources over VBR. In this effort, we also briefly survey VBR video modeling techniques, the MPEG-2 over ATM approach, and propose a model for MPEG-2 video over VBR which incorporates the long-range dependence property in compressed video. Compressed video sources exhibit long-range dependence in the traffic patterns they generate. We verify that the ERICA+ algorithm is robust to the variation introduced by such background traffic and can control the ABR queues.

Finally, in section 8.19, we refer to a related study of the effect of bursty applications (such as the World Wide Web application) running on top of TCP. Bursty applications can potentially cause unbounded ABR queues since they can use open TCP windows to send bursts of data. However, Vandalore et al [115] show that since ABR switches respond to load increases, if the aggregate load increases as a function of the number of applications, then the switch will assign lower rates to sources and

hence control the total load on the network. In other words, a well-designed ABR system can scale well to support a large number of applications like persistant file transfer or bursty WWW sources running over TCP.

BIBLIOGRAPHY

[1] Santosh P. Abraham and Anurag Kumar. Max-Min Fair Rate Control of ABR Connections with Nonzero MCRs. *IISc Technical Report*, 1997.

[2] Yehuda Afek, Yishay Mansour, and Zvi Ostfeld. Convergence Complexity of Optimistic Rate Based Flow Control Algorithms. In *28th Annual Symposium on Theory of Computing (STOC)*, pages 89–98, 1996.

[3] Yehuda Afek, Yishay Mansour, and Zvi Ostfeld. Phantom: A Simple and Effective Flow Control Scheme. In *Proceedings of the ACM SIGCOMM*, pages 169–182, August 1996.

[4] Anthony Alles. ATM Internetworking. White paper, Cisco Systems, http://www.cisco.com, May 1995.

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

[6] Ambalavanar Arulambalam, Xiaoqiang Chen, and Nirwan Ansari. Allocating Fair Rates for Available Bit Rate Service in ATM Networks. *IEEE Communications Magazine*, 34(11):92–100, November 1996.

[7] A.W.Barnhart. Changes Required to the Specification of Source Behavior. ATM Forum 95-0193, February 1995.

[8] A.W.Barnhart. Evaluation and Proposed Solutions for Source Behavior # 5. ATM Forum 95-1614, December 1995.

[9] A. W. Barnhart. Use of the Extended PRCA with Various Switch Mechanisms. ATM Forum 94-0898, 1994.

[10] A. W. Barnhart. Example Switch Algorithm for TM Spec. ATM Forum 95-0195, February 1995.

[11] J. Bennett, K. Fendick, K.K. Ramakrishnan, and F. Bonomi. RPC Behavior as it Relates to Source Behavior 5. ATM Forum 95-0568R1, May 1995.

[12] J. Bennett and G. Tom Des Jardins. Comments on the July PRCA Rate Control Baseline. ATM Forum 94-0682, July 1994.

[13] J. Beran, R. Sherman, M. Taqqu, and W. Willinger. Long-Range Dependence in Variable-Bit-Rate Video Traffic. *IEEE Transactions on Communications*, 43(2/3/4), February/March/April 1995.

[14] U. Black. *ATM: Foundation for Broadband Networks*. Prentice Hall, New York, 1995.

[15] P. E. Boyer and D. P. Tranchier. A reservation principle with applications to the atm traffic control. *Computer Networks and ISDN Systems*, 1992.

[16] D. Cavendish, S. Mascolo, and M. Gerla. SP-EPRCA: an ATM Rate Based Congestion Control Scheme basedon a Smith Predictor. Technical report, UCLA, 1997.

[17] Y. Chang, N. Golmie, L. Benmohamed, and D. Siu. Simulation study of the new rate-based eprca traffic management mechanism. *ATM Forum 94-0809*, 1994.

[18] A. Charny, G. Leeb, and M. Clarke. Some Observations on Source Behavior 5 of the Traffic Management Specification. ATM Forum 95-0976R1, August 1995.

[19] Anna Charny. An Algorithm for Rate Allocation in a Cell-Switching Network with Feedback. Master's thesis, Massachusetts Institute of Technology, May 1994.

[20] Anna Charny, David D. Clark, and Raj Jain. Congestion control with explicit rate indication. In *Proceedings of the IEEE International Communications Conference (ICC)*, June 1995.

[21] D. Chiu and R. Jain. Analysis of the Increase/Decrease Algorithms for Congestion Avoidance in Computer Networks. *Journal of Computer Networks and ISDN Systems*, 1989.

[22] Fabio M. Chiussi, Ye Xia, and Vijay P. Kumar. Dynamic max rate control algorithm for available bit rate service in atm networks. In *Proceedings of the IEEE GLOBECOM*, volume 3, pages 2108–2117, November 1996.

[23] D.P.Heyman and T.V. Lakshman. What are the implications of Long-Range Dependence for VBR-Video Traffic Engineering ? *ACM/IEEE Transactions on Networking*, 4(3):101–113, June 1996.

[24] Harry J.R. Dutton and Peter Lenhard. *Asynchronous Transfer Mode (ATM) Technical Overview*. Prentice Hall, New York, 2nd edition, 1995.

[25] H. Eriksson. MBONE: the multicast backbone. *Communications of the ACM*, 37(8):54–60, August 1994.

[26] J. Scott et al. Link by Link, Per VC Credit Based Flow Control. ATM Forum 94-0168, 1994.

[27] L. Roberts et al. New pseudocode for explicit rate plus efci support. *ATM Forum 94-0974*, 1994.

[28] M. Hluchyj et al. Closed-loop rate-based traffic management. *ATM Forum 94-0438R2*, 1994.

[29] M. Hluchyj et al. Closed-Loop Rate-Based Traffic Management. ATM Forum 94-0211R3, April 1994.

[30] S. Fahmy, R. Jain, S. Kalyanaraman, R. Goyal, and F. Lu. On source rules for abr service on atm networks with satellite links. In *Proceedings of First International Workshop on Satellite-based Information Services (WOSBIS)*, November 1996.

[31] Chien Fang and Arthur Lin. A Simulation Study of ABR Robustness with Binary-Mode Switches: Part II. ATM Forum 95-1328R1, October 1995.

[32] Chien Fang and Arthur Lin. On TCP Performance of UBR with EPD and UBR-EPD with a Fair Buffer Allocation Scheme. ATM Forum 95-1645, December 1995.

[33] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, and T. Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. Request For Comments, RFC 2068, January 1997.

[34] ATM Forum. http://www.atmforum.com.

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

[36] M. Garrett and W. Willinger. Analysis, modeling, and generation of self-similar vbr video traffic. In *Proceedings of the ACM SIGCOMM*, August 1994.

[37] Matthew S. Goldman. Variable Bit Rate MPEG-2 over ATM: Definitions and Recommendations. ATM Forum 96-1433, October 1996.

[38] Rohit Goyal, Raj Jain, Shiv Kalyanaraman, Sonia Fahmy, and Seong-Cheol Kim. Performance of TCP over UBR+. ATM Forum 96-1269, October 1996.

[39] Rohit Goyal, Raj Jain, Shiv Kalyanaraman, Sonia Fahmy, Bobby Vandalore, Xiangrong Cai, and Seong-Cheol Kim. Selective Acknowledgements and UBR+ Drop Policies to Improve TCP/UBR Performance over Terrestrial and Satellite Networks. ATM Forum 97-0423, April 1997.

[40] Rohit Goyal, Raj Jain, Shiv Kalyanaraman, Sonia Fahmy, Bobby Vandalore, Xiangrong Cai, and Seong-Cheol Kim. Selective Acknowledgements and UBR+ Drop Policies to Improve TCP/UBR Performance over Terrestrial and Satellite Networks. ATM Forum 97-0423, April 1997.

[41] M. Grossglauser, S.Keshav, and D.Tse. RCBR: a simple and efficient service for multiple time-scale traffic. In *Proceedings of the ACM SIGCOMM*, August 1995.

[42] S. Hrastar, H. Uzunalioglu, and W. Yen. Synchronization and de-jitter of mpeg-2 transport streams encapsulated in aal5/atm. In *Proceedings of the IEEE International Communications Conference (ICC)*, volume 3, pages 1411–1415, June 1996.

[43] D. Hughes and P. Daley. More abr simulation results. *ATM Forum 94-0777*, 1994.

[44] D. Hunt, Shirish Sathaye, and K. Brinkerhoff. The realities of flow control for abr service. *ATM Forum 94-0871*, 1994.

[45] Van Jacobson. Congestion avoidance and control. In *Proceedings of the ACM SIGCOMM*, pages 314–329, August 1988.

[46] J. Jaffe. Bottleneck Flow Control. *IEEE Transactions on Communications*, COM-29(7):954–962, 1980.

[47] R. Jain. A delay-based approach for congestion avoidance in interconnected heterogeneous computer networks. *Computer Communications Review*, 19.

[48] R. Jain. A timeout-based congestion control scheme for window flow-controlled networks. *IEEE Journal on Selected Areas in Communications*, 1986.

[49] R. Jain. A comparison of hashing schemes for address lookup in computer networks. *IEEE Transactions on Communications*, 1992.

[50] R. Jain. The eprca+ scheme. *ATM Forum 94-0988*, 1994.

[51] R. Jain. The osu scheme for congestion avoidance using explicit rate indication. *ATM Forum 94-0883*, 1994.

[52] R. Jain. Atm networking: Issues and challenges ahead. *Engineers Conference, InterOp+Network World*, 1995.

[53] R. Jain. Congestion control and traffic management in atm networks: Recent advances and a survey. *Computer Networks and ISDN Systems*, 1995.

[54] R. Jain, D. Chiu, and W. Hawe. A quantitative measure of fairness and discrimination for resource allocation in shared systems. *DEC TR-301*, 1984.

[55] R. Jain, D. Chiu, and W. Hawe. A quantitative measure of fairness and discrimination for resource allocation in shared systems. *DEC TR-301*, 1984.

[56] R. Jain, S. Kalyanaraman, R. Goyal, S. Fahmy, and F. Lu. A Fix for Source End System Rule 5. ATM Forum 95-1660, December 1995.

[57] R. Jain, S. Kalyanaraman, R. Goyal, S. Fahmy, and F. Lu. Erica+: Extensions to the erica switch algorithm. *ATM Forum 95-1145R1*, 1995.

[58] R. Jain, S. Kalyanaraman, and R. Viswanathan. Method and apparatus for congestion management in computer networks using explicit rate indication. *U. S. Patent application filed (S/N 307, 375),*, 1994.

[59] R. Jain, S. Kalyanaraman, and R. Viswanathan. The transient performance: Eprca vs eprca++. *ATM Forum 94-1173*, 1994.

[60] R. Jain, S. Kalyanaraman, R. Viswanathan, and R. Goyal. A sample switch algorithm. *ATM Forum 95-0178R1*, 1995.

[61] R. Jain, K. Ramakrishnan, and D Chiu. Congestion avoidance scheme for computer networks. *U.S. Patent #5377322,*, 1994.

[62] R. Jain and K. K. Ramakrishnan. Congestion avoidance in computer networks with a connectionless network layer: Concepts, goals, and methodology. *Proc. IEEE Computer Networking Symposium*, 1988.

[63] R. Jain, K. K. Ramakrishnan, and D. M. Chiu. Congestion Avoidance in Computer Networks with a Connectionless Network Layer. Technical Report DEC-TR-506, Digital Equipment Corporation, August 1987.

[64] R. Jain and S. Routhier. Packet Trains - Measurement and a new model for computer network trafic. *IEEE Journal of Selected Areas in Communications,*, 1986.

[65] Raj Jain. Congestion Control in Computer Networks: Issues and Trends. *IEEE Network Magazine*, pages 24–30, May 1990.

[66] Raj Jain. *The Art of Computer Systems Performance Analysis.* John Wiley & Sons, 1991.

[67] Raj Jain. Myths about Congestion Management in High-speed Networks. *Internetworking: Research and Experience*, 3:101–113, 1992.

[68] Raj Jain. ABR Service on ATM Networks: What is it? Network World, 1995.

[69] Raj Jain. Congestion Control and Traffic Management in ATM Networks: Recent advances and a survey. *Computer Networks and ISDN Systems Journal*, October 1996.

[70] Raj Jain, Sonia Fahmy, Shivkumar Kalyanaraman, Rohit Goyal, and Fang Lu. More Straw-Vote Comments: TBE vs Queue sizes. ATM Forum 95-1661, December 1995.

[71] Raj Jain, Shiv Kalyanaraman, Rohit GOyal, and Sonia Fahmy. Source Behavior for ATM ABR Traffic Management: An Explanation. *IEEE Communications Magazine*, 34(11), November 1996.

[72] Raj Jain, Shivkumar Kalyanaraman, Sonia Fahmy, and Fang Lu. Bursty ABR Sources. ATM Forum 95-1345, October 1995.

[73] Raj Jain, Shivkumar Kalyanaraman, Sonia Fahmy, and Fang Lu. Out-of-Rate RM Cell Issues and Effect of Trm, TOF, and TCR. ATM Forum 95-973R1, August 1995.

[74] Raj Jain, Shivkumar Kalyanaraman, Sonia Fahmy, and Fang Lu. Straw-Vote comments on TM 4.0 R8. ATM Forum 95-1343, October 1995.

[75] Raj Jain, Shivkumar Kalyanaraman, Rohit Goyal, Ram Viswanathan, and Sonia Fahmy. Erica: Explicit rate indication for congestion avoidance in atm networks. U.S. Patent Application (S/N 08/683,871), July 1996.

[76] Raj Jain, Shivkumar Kalyanaraman, and Ram Viswanathan. The osu scheme for congestion avoidance in atm networks: Lessons learnt and extensions. *Performance Evaluation Journal*, October 1997. to appear.

[77] Raj Jain and Shivkumar Kalyanaraman Ram Viswanathan. 'method and apparatus for congestion management in computer networks using explicit rate indication. U. S. Patent application (S/N 307,375), SepJuly 1994.

[78] H. Tzeng K. Siu. Intelligent congestion control for abr service in atm networks. *Computer Communication Review*, 24(5):81–106, October 1995.

[79] Lampros Kalampoukas, Anujan Varma, and K.K. Ramakrishnan. An efficient rate allocation algorithm for atm networks providing max-min fairness. In *6th IFIP International Conference on High Performance Networking (HPN)*, September 1995.

[80] Shivkumar Kalyanaraman, Raj Jain, Sonia Fahmy, Rohit Goyal, and Jianping Jiang. Performance of TCP over ABR on ATM backbone and with various VBR traffic patterns. In *Proceedings of the IEEE International Communications Conference (ICC)*, June 1997.

[81] Shivkumar Kalyanaraman, Raj Jain, Rohit Goyal, and Sonia Fahmy. A Survey of the Use-It-Or-Lose-It Policies for the ABR Service in ATM Networks. Technical Report OSU-CISRC-1/97-TR02, Dept of CIS, The Ohio State University, 1997.

[82] D. Kataria. Comments on rate-based proposal. *ATM Forum 94-0384*, 1994.

[83] J.B. Kenney. Problems and Suggested Solutions in Core Behavior. ATM Forum 95-0564R1, May 1995.

[84] Bo-Kyoung Kim, Byung G. Kim, and Ilyoung Chong. Dynamic Averaging Interval Algorithm for ERICA ABR Control Scheme. ATM Forum 96-0062, February 1996.

[85] H. T. Kung. Adaptive Credit Allocation for Flow-Controlled VCs. ATM Forum 94-0282, 1994.

[86] H. T. Kung. Flow Controlled Virtual Connections Proposal for ATM Traffic Management. ATM Forum 94-0632R2, September 1994.

[87] T.V. Lakshman, P.P. Mishra, and K.K. Ramakrishnan. Transporting compressed video over atm networks with explicit rate feedback control. In *Proceedings of the IEEE INFOCOM*, April 1997.

[88] L.G.Roberts. Operation of Source Behavior # 5. ATM Forum 95-1641, December 1995.

[89] Hongqing Li, Kai-Yeung Siu, Hong-Ti Tzeng, Chinatsu Ikeda, and Hiroshi Suzuki. Tcp over abr and ubr services in atm. In *Proceedings of IPCCC'96*, March 1996.

[90] S. Liu, M. Procanik, T. Chen, V.K. Samalam, and J. Ormond. An analysis of source rule # 5. ATM Forum 95-1545, December 1995.

[91] B. Lyles and A. Lin. Definition and preliminary simulation f a rate-based congestion control mechanism with explicit feedback of bottleneck rates. *ATM Forum 94-0708*, 1994.

[92] P. Newman. Traffic Management for ATM Local Area Networks. *IEEE Communications Magazine*, 1994.

[93] P. Newman and G. Marshall. Becn congestion control. *ATM Forum 94-789R1*, 1993.

[94] P. Newman and G. Marshall. Update on becn congestion control. *ATM Forum 94-855R1*, 1993.

[95] Craig Partridge. *Gigabit Networking*. Addison-Wesley, Reading, MA, 1993.

[96] Vern Paxson. Fast Approximation of Self-Similar Network Traffic. Technical Report LBL-36750, Lawrence Berkeley Labs, April 1995.

[97] K. Ramakrishnan and R. Jain. A binary feedback scheme for congestion avoidance in computer networks with connectionless network layer. *ACM Transactions on Computers*, 1990.

[98] K. K. Ramakrishnan, D. M. Chiu, and R. Jain. Congestion Avoidance in Computer Networks with a Connectionless Network Layer. Part IV: A Selective Binary Feedback Scheme for General Topologies. Technical report, Digital Equipment Corporation, 1987.

[99] K. K. Ramakrishnan and P. Newman. Credit where credit is due. *ATM Forum 94-0916*, 1994.

[100] K. K. Ramakrishnan and "Issues with Backward Explicit Congestion Notification based Congestion Control. Issues with backward explicit congestion notification based congestion control. *ATM Forum 94-0231*, 1993.

[101] K. K. Ramakrishnan and J. Zavgren. Preliminary simulation results of hop-by-hop/vc flow control and early packet discard. *ATM Forum 94-0231*, 1994.

[102] K.K. Ramakrishnan, P. P. Mishra, and K. W. Fendick. Examination of Alternative Mechanisms for Use-it-or-Lose-it. ATM Forum 95-1599, December 1995.

[103] L. Roberts. The benefits of rate-based flow control for abr service. *ATM Forum 94-0796*, 1994.

[104] L. Roberts. Enhanced prca (proportional rate-control algorithm). *ATM Forum 94-0735R1*, 1994.

[105] L. Roberts. Rate-based algorithm for point to multipoint abr service. *ATM Forum 94-0772R1*, 1994.

[106] Larry Roberts. Enhanced PRCA (Proportional Rate-Control Algorithm). ATM Forum 94-0735R1, August 1994.

[107] A. Romanov. A performance enhancement for packetized abr and vbr+ data. *ATM Forum 94-0295*, 1994.

[108] Allyn Romanov and Sally Floyd. Dynamics of TCP Traffic over ATM Networks. *IEEE Journal on Selected Areas in Communications*, May 1995.

[109] W. Stallings. Isdn and broadband isdn with frame relay and atm. *ATM Forum 94-0888*, 1995.

[110] Lucent Technologies. Atlanta chip set, microelectronics group news announcement, http://www.lucent.com/micro/news/032497.html.

[111] Christos Tryfonas. MPEG-2 Transport over ATM Networks. Master's thesis, University of California at Santa Cruz, September 1996.

[112] H. Tzeng and K. Siu. A class of proportional rate control schemes and simulation results. *ATM Forum 94-0888*, 1994.

[113] H. Tzeng and K. Siu. Enhanced credit-based congestion notification (eccn) flow control for atm networks. *ATM Forum 94-0450*, 1994.

[114] International Telecommunications Union. http://www.itu.ch.

[115] Bobby Vandalore, Shiv Kalyanaraman, Raj Jain, Rohit Goyal, Sonia Fahmy, Xiangrong Cai, and Seong-Cheol Kim. Performance of Bursty World Wide Web (WWW) Sources over ABR. ATM Forum 97-0425, April 1997.

[116] Bobby Vandalore, Shiv Kalyanaraman, Raj Jain, Rohit Goyal, Sonia Fahmy, and Pradeep Samudra. Worst case TCP behavior over ABR and buffer requirements. ATM Forum 97-0617, July 1997.

[117] L. Wojnaroski. Baseline text for traffic management sub-working group. *ATM Forum 94-0394R4*, 1994.

[118] Gary R. Wright and W. Richard Stevens. *TCP/IP Illustrated, Volume 2.* Addison-Wesley, Reading, MA, 1995.

[119] Lixia Zhang, Scott Shenker, and D.D.Clark. Observations on the dynamics of a congestion control algorithm: The effects of two-way traffic. In *Proceedings of the ACM SIGCOMM*, August 1991.