Abstract

This work compares two ATM services that can be used as the underlying infrasructure to support data traffic that employ TCP and UDP protocols. These two services are the Available Bit Rate (ABR) service and the Unspecified Bit Rate with Early Packet Discard (UBR+EPD). The performance assessment is made in two network scenarios: 1) the network is end-to-end ATM i.e., the host is directly connected to the ATM network and 2) IP/ATM internetworks. Simulation is used to estimate the performance of these networks. End-to-End throughput/efficiency and fairness index are used here as the key performance metrics in comparing the two service classes.

It was found that in all the end-to-end ATM scenarios considered here, ABR always outperformed UBR+EPD based on both efficiency and fairness. In the non-end-to-end ATM scenarios, for single congested node cases the performance with both ABR and UBR+EPD was similar while ABR gave marked performance gains over UBR+EPD as the number of congested nodes/links increased. However, the performance gain in the non-end-to-end ATM cases was lesser in magnitude compared to the end-to-end ATM cases. Service providers can use this work as a reference in seeing the gains attainable by deploying ABR.

The University of Kansas

Information and Telecommunication Technology Center

Technical Report

Performance Analysis of TCP/IP and UDP/IP Over ATM Networks: ABR Vs UBR+EPD Services

Haritha Pindi & Victor Frost

ITTC-FY99-TR-13200-06

September 1998

Project Sponsor:
Sprint Corporation

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Broadband networks of the future will not only integrate the current services and bring a variety of new applications such as medical imaging, corporation intranets and video on demand but will also facilitate the development of a unified management system for all services. Asynchronous Transfer Mode (ATM) is the technology selected to support this integrated services network [3]. At the same time, the Internet, has grown spectacularly and it is supporting the vast majority of the present data traffic. The success of the Internet family of protocols suggests they will continue to be a significant portion of the traditional data traffic. In order for ATM to succeed, it must support efficiently the legacy applications of the Internet. This can only be realized with effective traffic management mechanisms and through proper internetworking between TCP/IP and ATM.

Until recently, virtually all of the experience in running TCP and IP has been over networks with relatively few features for congestion control and quality of service, such as X.25 WANs and IEEE 802 LANS [4]. Increasingly however, the TCP/IP protocol stack is being used over ATM Networks. The growing use of TCP/IP over ATM has sparked research on its performance. The default TCP/IP performance over a congested and uncontrolled ATM network is poor. A number of simulation studies [6, 7, 9] have predicted this behavior and experimental studies [15, 16, 17] have confirmed the same.

IP is a packet switched technology originally designed for connectionless, shared medium networks, whereas ATM is a cell switched, connection-oriented technology.

Since TCP was designed to work independently of the lower layer implementations, it seems that it should pose no problems with ATM as well. However, this very fundamental assumption that TCP makes of the underlying network is not valid for ATM networks. The disparity in properties of the underlying ATM network is the principal cause for poor performance of TCP over congested ATM networks. TCP packets get fragmented into cells for transport over ATM and a single cell drop at the ATM level results in the entire packet being dropped at the TCP level. TCP is forced to retransmit the entire packet and these retransmissions drastically affect throughput. Secondly, useful bandwidth and also the buffer space get wasted as the congested link transmits cells from corrupted packets (packets in which at least one cell has been dropped). Partial Packet and Early Packet Discard have been proposed by Romanow [6] to alleviate the problem of fragmentation loss and corrupted packets. This has improved the situation to some extent. There were several other research studies that came up with a variety of suggestions to this problem of improving TCP throughput over ATM netowrks. Cell-level traffic shaping schemes have been suggested in [22]. Cell level pacing is the mechanism which reduces the source cell transmission rate. The experimental results in [15] show that TCP rate control mechanism is not effective in controlling the traffic burstiness sufficiently to avoid congestion induced cell losses in wide area ATM networks. It was illustrated that TCP augmented with cell level pacing improves performance and allows full utilization of the bandwidth capacity [15]. This kind of traffic shaping occurs on the host side as opposed to the policing mechanism (rate control) which occurs inside the network. Packet based traffic shaping and an aggressive feedback retransmission policy were proposed in [5]. It was shown that providing a traffic shaping envelope at the TCP layer is instrumental in achieving the goal of QoS guarantees to the application layer. This, however, requires changes to the TCP protocol which might not be a feasible solution. It will be important to have schemes that can improve performance without changes to the TCP protocol so that the existing applications can take advantage of it. *Are there any such schemes?* The answer is yes. ATM networks are capable of complex quality-of-service functions and have a wide variety of congestion and traffic control facilities. These ATM level congestion control schemes could be used to improve the TCP performance.

Further, the solutions that were devised so far aimed at improving only the TCP throughput and addressed the problem when there is no ATM-level congestion control. While throughput is definitely a very important performance measure, achieving fairness to all the contending users, ensuring efficient utilization of the link capacities and the buffers in the network, low average delay are also very important and the ATM layer congestion control policies could be effectively used to meet these additional goals. The problem now takes a different dimension in the sense that *What are the ATM layer congestion control features that can help meet these goals?*. Obviously, these congestion control features have to be intended for data traffic. ATM networks provide five classes of service [10] - Constant Bit Rate (CBR), real-time Variable Bit Rate(rt_VBR), non-real time Variable Bit Rate (nrt_VBR), Available Bit Rate (ABR) and Unspecified Bit Rate (UBR). Of these, ABR and UBR are intended mainly for data traffic and hence are the candidates for transporting TCP/IP or UDP/IP traffic. Now, *do these service classes have any such congestion control features that we are interested in?*. Yes, the ABR service has a *rate-based closed-loop control* feature that requires the network switches to constantly monitor their load and feed the congestion information back to the sources, which in turn dynamically adjust their input rate into the network. While the UBR service does not have any explicit congestion control features, it does require the switches to monitor their queues and simply discard cells or packets of overloading users. So, either of these services could be used for data traffic. Also it should be mentioned here that ABR is intended to minimize cell losses in the network, while UBR makes no such attempt. The issue now remains *which of these two service classes is the most suitable one for TCP?*. Actually, UBR is nothing but TCP over ATM without any congestion control features and it is already proved to give poor congestion performance. However, UBR enhanced with early packet discard (UBR+EPD) provides significant improvement in comparison to UBR service with simple packet discard [6]. So we need a comparison between ABR and UBR+EPD services for data traffic.

In essence, the issue is how best to manage TCP's segment size, window management, and congestion control policies on the one hand, and ATM's quality-of-service and traffic policies on the other, to achieve high throughput for TCP traffic, fair allocation among various TCP connections, and efficient use of the underlying ATM network. This issue is complex because of the many factors involved. Further, the context in which TCP operates over ATM is a factor. TCP/IP can operate end-to-end over a single ATM network, or there may be one or more ATM LANS and/or ATM WANS along an internetwork route that also involve non-ATM networks.

1.1 Problem Definition and Proposed Work

Given that the ATM networks do have congestion control facilities and these being incorporated in the ABR and UBR service classes intended for data traffic, the next question that arises is

Which of these service classes - ABR and UBR+EPD, is the best class for TCP?

UBR has been the preferred service for TCP traffic since it is simple and added advantage of EPD schemes is that they are relatively inexpensive to be implemented in ATM switches. ABR service is relatively new and is considerably more complex in terms of hardware compared to UBR as it requires an algorithm to be implemented in the switch and also requires special features on the host side. However ABR is increasingly implemented by ATM switch vendors and hence it is worthwhile to look at the performance of TCP over ABR to determine *if there is any performance improvement compared to UBR, how much is the degree of improvement in performance?, and is it worth the complexity and the extra cost?.*

One may expect that since ABR attempts to avoid loss in the ATM network, the performance of TCP over ABR will be better than that over UBR. However, this is not necessarily the case since there are several factors that may lead to degradation of TCP performance over ABR. These factors include

- Traffic overhead of Resource Management (RM) cells
- ABR introduces extra queueing stage at the edge of the ATM network (a cell buffer after the AAL5 module in the ATM host network interface cards), which can increase end-to-end delay. While with UBR no such queueing delays are

experienced at the edge as all traffic is sent out at peak cell rate which is usually equal to the line rate. This holds true particularly in the case of end-to-end ATM networks. However, since most ABR algorithms attempt to keep the ABR buffer fill to a relatively small value, the queueing delay in the switches could be small and compensate for the delays incurred at the edge. With UBR, on the other hand, the queueing occurs in the switches along the path. So, the comparison primarily lies between the queueing delay experienced at the switches in the case of UBR+EPD and at the edge controllers in the case of ABR. This issue of potential additional delays needs clarification and hence an investigation into the delay behavior is an important issue.

- The ABR control loop is designed to slow down sources in case of congestion; however in an internetworking environment (i.e., in the case of non end-to-end ATM networks), the source that ABR controls is simply a router and there is no signalling (at least the current standards do not have such a provision) of feedback to the real traffic sources (workstations, PCs etc) the rate at which they should transmit. This means congestion may be avoided in the ATM part of the network but not for the end-to-end TCP flows, i.e, there are no performance gains as of the application is concerned.

- The introduction of ABR control loop inside the TCP congestion loop may lead to undesirable interactions between the two that may degrade performance.

- Finally, for ABR to properly achieve its objectives of high link utilization and of low loss, many parameters may have to be correctly tuned.

So it is not certain that ABR is the correct service class for TCP applications. Hence, an extensive study comparing the performance of the two services classes is needed. Also, though majority of the internet traffic uses TCP for the transport protocol, significant portion of the traffic especially voice and video applications developed for the Internet are based upon UDP. Examples of applications that use UDP are the Real-Time Protocol (RTP), SNMP etc. TCP with ATM flow control and with its own implicit congestion handling mechanism can keep up its throughput in the event of congestion. UDP based applications unlike TCP based applications do not back off or reduce their input rate in the case of network congestion and interfere with other traffic. Whether ABR is more favorable to UDP than to TCP (relative to UBR) is a research issue in itself.

This research conducts a study of the performance of data traffic sources employing TCP and UDP protocols over ATM networks with ABR service and UBR service with Early Packet Discard Schemes. The EPD mechanism proposed in [6] is used in the studies. For the ABR congestion control schemes, an Explicit Rate algorithm is chosen. There will be no significant difference in the simulation results if we choose to use other equally promising explicit rate schemes proposed in the ATM Forum [10]. The motivation for our work is to understand the degree of improvement in TCP performance using a more expensive ABR congestion control in comparison with a simpler UBR service. There are a wide variety of parameters *viz.,* topology, algorithm parameters, feedback delay, round-trip times, TCP parameters etc., that may have a significant impact on the performance of the ABR and UBR implementations which make it hard to pursue a thorough simulation study. We devise our simulation framework to take the **load, the number of congested links and the location of the point of congestion** as the parameter space for performance assessment.

We focus our study on a wide area networks(WAN) using two-node and multiple-node network configurations and consider the two cases- 1) the network is homogeneous i.e., end-to-end ATM and 2) heterogeneous networks consisting of an ATM cloud that connects various TCP/IP subnetworks (non end-to-end ATM). In summary, the goal is to make

- *Qualitative and Quantitative comparison of TCP performance over the ABR and UBR+EPD service classes in various WAN configurations for both end-to-end ATM networks and non-end-to-end ATM networks.*

- *Identify the performance tradeoffs between the two service classes.*

- *Provide guidelines to service providers based on the identified tradeoffs.*

Our performance measures are throughput, efficiency, fairness index, and gain (if any) in using ABR over UBR+EPD.

1.2 Previous Work and Motivation

The problem of evaluating TCP performance over the two ATM service classes - ABR and UBR has drawn the attention of several researchers [8, 20, 21], but, with few consensus conclusions so far. Most of the research findings may be considered premature for the following reasons.

1. Majority of the research was too simplistic to draw general conclusions. For example, the performance is highly dependent on network topology considered in each of these studies. Most of the studies so far has been on the two node topology with conclusions stemming henceforth that one service is categorically better than the other. It is not appropriate to draw general conclusions without examining more complex topologies that include multiple nodes, multiple bottlenecks and multiple classes of traffic. In our work, in addition to the simple two node configuration, we also investigate multiple node configurations and consider more realistic scenarios where background traffic is present.

2. The specific system parameters also vary from study-to-study. It is not surprising that two different studies using the same topology can come up with opposing conclusions as they use different parameters for buffer sizes, loading levels, the version of TCP and the particular ABR explicit rate flow control algorithm.

3. Most of the reported studies [8, 20, 21] have assumed a simple switch model with output buffering and FIFO queueing. We believe that the switch architecture and the scheduling mechanism used in the switches could have impact on the results. A realistic switch model that implements realistic scheduling mechanisms like the per-VC queueing and weighted round robin scheduling is used in this study.

4. There has been little work on end-to-end traffic management in the internetworking environment, where networks are heterogeneous i.e, TCP/IP subnetworks connected by ATM clouds. In such cases ABR flow control may simply push the congestion to the edge routers. Hence for a comparison between ABR and UBR service classes, such cases need to be considered.

5. Most of the work to-date [6, 18, 19] have reported only throughput as the performance measure. To obtain comprehensive results, performance measures like throughput, fairness index, cell loss rate, packet loss rate and link utilizations are addressed in our study.

1.3 Organization of the Thesis

The rest of this document is organized as follows:

- Chapter 2 provides background material on TCP Congestion Control features, Traffic Management and Congestion control in ATM networks and issues in internetworking TCP/IP with ATM and the performance implications.
- Chapter 3 describes the experimental scenarios and the rationale for the choice of the experiments.
- Chapter 4 describes the simulation set up, the network configuration set up, the simulation model, model assumptions, the simulation parameters and the rationale of their selection.
- Chapter 5 presents the results of the various experiments and a discussion of these results.
- Several conclusions from this research are discussed in Chapter 6. Directions for future work are also included here.

Chapter 2

Background

2.1 TCP Congestion Control

The TCP/IP protocol suite consists of several protocols for different purposes. The transmission control protocol (TCP) is the primary transport protocol in the TCP/IP protocol suite. Since our focus is on the congestion control for TCP over ATM, only the features connected with the TCP end system flow control mechanisms are discussed here.

TCP is a reliable, connection-oriented, full-duplex, end-to-end transport protocol [11] that provides flow controlled byte stream service. The only tool in TCP that relates to network congestion is the sliding-window flow and error control mechanism. TCP does not have an explicit mechanism for congestion control[1, 2]. It only provides end-to-end flow control: ensuring that data is transmitted at a rate consistent with the capacities of both the receiver and the intermediate links in the network path. The TCP implicit congestion control mechanism is primarily based on the dynamic-window adjustment and retransmission timeouts. A sliding window protocol is used for flow control which allows the sender to transmit multiple packets before it stops and waits for an acknowledgment (ACK). The sender keeps a record of each packet it sends until it receives the ACKs. The sender also starts a retransmission timer when it sends a packet. Upon receiving an ACK, the sender sends more packets within allowed window size. Timeout occurs when no ACK is received, and the corresponding packet is retransmitted. A timeout may occur due to a variety of reasons such as lost packet or lost ACK or simply due to excessive delay in the network which may occur under severe network congestion. Here lost segments imply congestion and TCP takes actions for adjusting the flow into the network. Although the sliding window mechanism is designed for end-to-end flow control, a number of techniques have been developed and incorporated into different versions of TCP implementations that enable it to be used for congestion detection, avoidance and recovery. We briefly survey below some of the most important and widely implemented mechanisms/algorithms.

Slow Start

As mentioned before, TCP uses a window based protocol for flow control. The sender maintains a variable congestion window (CWND) to control how many segments can be put into the network at a specific time, while the receiver maintains a receiver window (RCVWND)to tell the sender how many segments it can afford. Initially the congestion window is set to one segment and increased by one segment whenever it receives a new acknowledgment from the receiver until it reaches the minimum of RCWND and maximum CWND (i.e min(RCWND, maxCWND), which is normally 65535 bytes) ,i.e, when sending, send the minimum of the receiver's advertised window (RCWND) and CWND. This is called the slow start stage of TCP. Unlike its name suggests, the slow-start algorithm causes the congestion window to increase exponentially. During the slow start stage the congestion window doubles every round trip time. Note that the congestion window is flow control imposed by the sender, while the advertised window is flow control imposed by the receiver [2].

Congestion Avoidance

The Congestion Avoidance algorithm [14] deals with lost packets and has two components. Packet loss is detected either by retransmission timeout or duplicate ACKs. After detecting a packet loss (which is a likely symptom of network congestion), one-half of the current window size (the minimum of CWND and the receiver's advertised window, RCWND) but at least two segments is saved in SSTHRESH (Slow Start Threshold)

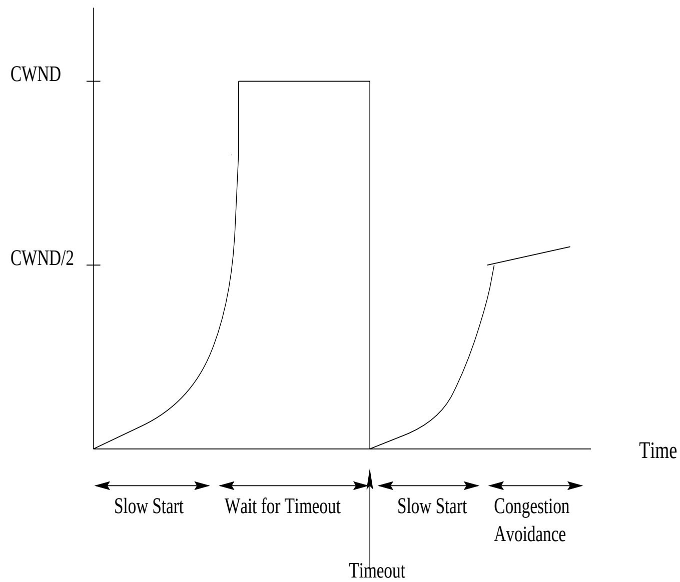


Figure 2.1: TCP Slow Start and Congestion Avoidance

and the CWND is reset to one (i.e slow-start). The sender then retransmits segments starting from the lost one. When this new data is acknowledged by the receiver, the CWND is increased by one packet for every ACK received until it is less than or equal to SSTHRESH. After that the CWND is incremented by 1/CWND each time an ACK is received (in other words CWND increases by one segment every round trip time). This results in a linear additive increase of CWND. This is called the congestion avoidance. Figure 2.1 shows the slow start and congestion avoidance stages for a typical TCP connection.

Fast Retransmit and Recovery

TCP uses a coarse granularity (typically 500ms) timer for retransmission timeout. As a result, the TCP connection can lose a lot of time waiting for the timeout when it suffers segment loss. During this waiting period, TCP neither transmits new packets nor retransmits the lost packets. Moreover, once the timeout occurs, the CWND is set to 1 segment, which means the connection will take few more round trip times to make full use of the network link. This will definitely result in low efficiency. As an improvement, fast retransmit and recovery (FRR) [12] was proposed to enable the connection recover from isolated segment loss quickly.

When the receiver receives an out of order segment, it sends a duplicate ACK to the sender immediately. The purpose of the duplicate ACK is to let the source know that a segment was received out-of-order, and to tell it what sequence number is expected. Since it is not certain whether a duplicate ACK is caused by a lost packet or just reordering of packets, the source waits for three duplicate ACKs to be received which is a strong indication that a packet has been lost. So, when the sender receives three duplicate ACKs, it assumes that the segment indicated by the ACKs is lost without waiting for the retransmission timer expiry. It then performs retransmission of the missing segment. This is the **fast retransmit** algorithm. New actions are taken. These actions are: the sender retransmits the lost segment immediately; the sender reduces the congestion window to half (plus 3 segments, which correspond to the 3 duplicate ACKs), and saves the original CWND value in SSTHRESH. Now for each subsequent duplicate ACK, the sender increases the CWND by one and tries to send a new segment. The effect of these actions is that the sender maintains the connection pipe at half of its capacity when it is in fast retransmit. For large propagation delay-bandwidth network environment, such fast retransmit mechanism can help avoiding unnecessary throughput degradation due to a large retransmission timeout interval. The algorithm first appeared in the 4.3BSD Tahoe release and the subsequent Net/1 release [2].

In 4.3BSD Tahoe version of TCP implementation, the fast retransmit algorithm is followed by slow start while the Reno version enters the congestion avoidance phase which is termed the fast recovery.

Without fast retransmit and recovery, every packet loss would cause the data pipeline to drain completely and would require a slow start action to recover. The fast retransmit and recovery algorithms have a combined effect of recovering from one packet loss per window, without draining the pipeline. However, more than one packet loss per window results in a retransmission timeout, pipeline drain and slow-start.

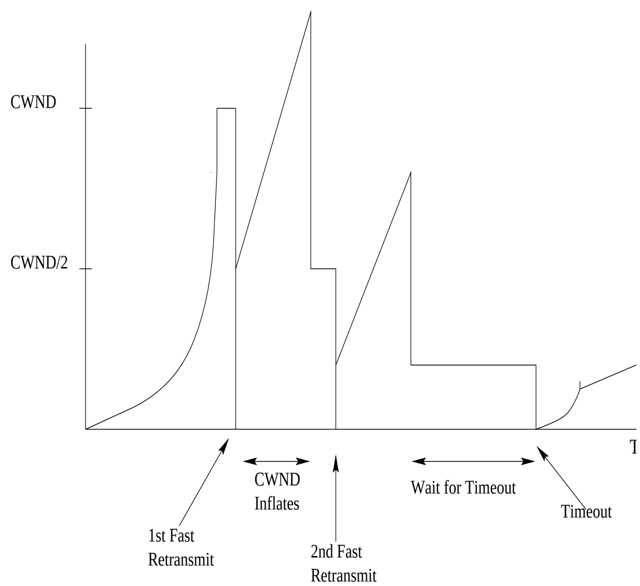


Figure 2.2: TCP Fast Retransmit and Recovery

Selective Acknowledgment

Fast Retransmit and Recovery works well for just isolated losses. If several losses occur in a short period of time, the performance of FRR is poor. A new proposal, Selective Acknowledgments (SACK) [13]works well even under multiple packet losses. A SACK TCP acknowledgment contains additional information about the segments that have been received by the destination. When duplicate SACKs are received from the destination, the sending TCP can reconstruct information about the segments not received at the destination. When the sender receives three duplicate ACKs, it retransmits the first lost segment, and increases the CWND by one for each duplicate ACK it receives. After that, whenever it is allowed to send another segment, it uses the SACK information to retransmit lost segments before sending any new segments. As a result, the sender can recover from multiple dropped segments in about one round trip time. Figure 2.3 shows the case of selective acknowledgment. SACK TCP is not yet implemented in any operating system.

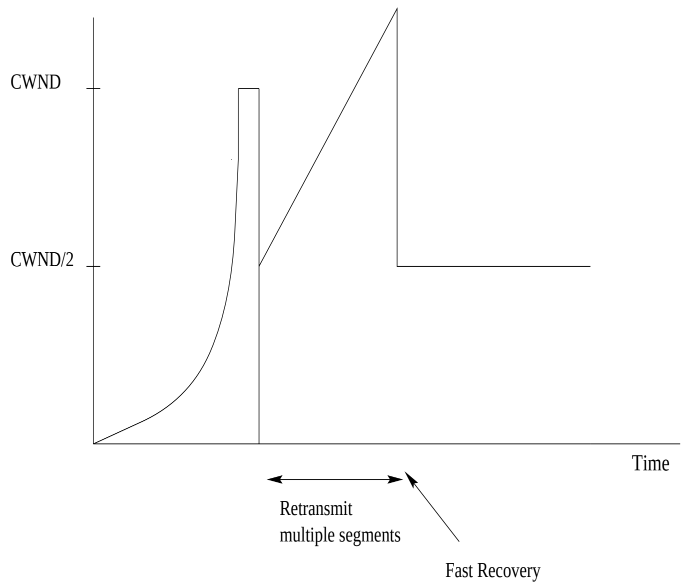


Figure 2.3: TCP Selective Acknowledgment

2.2 Traffic Management and Congestion Control in ATM Networks

ATM networks support multiple service classes, which include Constant Bit Rate (CBR), Variable Bit Rate (VBR), Available Bit Rate (ABR) and Unspecified Bit Rate (UBR) with different Quality-of-Service requirements. The control of ATM network traffic is fundamentally related to the ability of the network to provide appropriately differentiated Quality of Service (QoS) for network applications. A primary role of traffic management is to protect the network and end-system from congestion in order to achieve network performance objectives. An additional role is to promote the efficient use of network resources. In order for ATM networks to deliver guaranteed quality of service on demand while maximizing the utilization of available resources, effective traffic management techniques are needed. Almost every operation in an ATM network has some kind of traffic management mechanism.

Two types of Traffic Management Mechanisms have been proposed [10] for ATM networks: **Preventive** and **Reactive Control** mechanisms. The former corresponds to **Traffic Control** and the latter to **Congestion Control**. The purpose of **traffic**

control is to avoid/minimize congestion by regulating the different types of traffic being carried and ensuring that each type conforms to its expected behavior. The purpose of **congestion control** is to see the network can recover quickly and efficiently from congestion when it does occur. Connection Admission Control is one of the most important preventive control functions. To provide a guaranteed QOS, a traffic contract is established during connection setup. The traffic contract specifies the traffic class and the required QoS. CAC provides an algorithm which decides whether a new call can be accepted or not based on the availability of network resources to provide the acceptable QOS. Subsequently Usage Parameter Control/Network Parameter Control, a policing function is conducted to verify that the source adheres to the parameter values previously declared in the traffic contract. Other preventive functions such as Traffic Shaping, Network Resource Management (NRM) are defined [10]. In spite of these preventive measures, congestion may occur; reactive actions are necessary to minimize the intensity, spread and duration of the congestion. The ATM-level control structure has the capability of selectively discarding cells under congestion conditions. However, congestion control requires considerable exchange of information among the nodes of the network. This exchange occurs through the headers of the data cells and through some control cells.

The preventive mechanisms described above constitute the basics of traffic management and are adequate for voice and video communications (which constitute the CBR, rt_VBR and nrt_VBR service classes). However, they are obviously not sufficient for data communications. Data sources are extremely greedy and bursty. Further, the bandwidth requirements for data traffic is not likely to be known a priori at connection set up time. Thus some additional mechanism is needed for data traffic. Originally it was thought that data services could be classified as Unspecified Bit Rate (UBR) traffic, leaving hence the task to higher layer protocols. The latter should detect the cell losses due to the burstiness of the transmission and adjust then their sending rates accordingly. Nevertheless the lack of regulation at the ATM layer may result in a very high CLR (Cell Loss Ratio), which is unacceptable for data applications; at the same time, the QoS of connections sharing links with UBR traffic is also degraded: the CLR and CDV (Cell Delay Variation) are significantly increased. A new traffic category was defined: the Available Bit Rate (ABR) class. In ATM networks, the ABR service and UBR service are used to support non-delay sensitive data applications (e.g. file transfer, web access, remote procedure calls, distributed file service), and they are the two kinds of "best effort" service.

In the following subsections we will describe the features of ABR and UBR in the context of congestion control.

2.2.1 Available Bit Rate (ABR)

As the name suggests, Available Bit Rate normally uses the available bandwidth. This is often the left-over bandwidth of the higher priority services which are CBR and VBR. Though the current standards for ABR service do not require the cell transfer delay and cell loss ratio to be guaranteed, it is desirable for switches to minimize the delay and losses as much as possible. The ABR service requires the network switches to constantly monitor their load and feed this information back to the sources, which in turn dynamically adjust their input flow into the network. This is mainly done by inserting Resource Management (RM) cells into the traffic periodically and getting the network congestion state feedback from the returned RM cells, which may contain congestion information reported by the switches and destination. Depending upon the feedback, the source is required to adjust its transmission rate. Figure 2.4 shows an ABR traffic management model. An end-system that adapts its traffic in accordance with the feedback will experience a low cell loss ratio and obtain a fair share of the available bandwidth according to a network specific allocation policy.

The ATM Forum Traffic Management group [10] has standardized a **rate-based, closed-loop flow control** model for ABR service. The important features of this flow control model are presented here. On the establishment of an ABR connection, the end-system shall specify to the network both a maximum required bandwidth (Peak Cell Rate PCR) and a minimum usable bandwidth (Minimum cell rate, MCR). The MCR is guaranteed by the network to the VC. Most VCs use zero as the default MCR value. However, for an ABR with higher MCR, the connection may be denied if sufficient

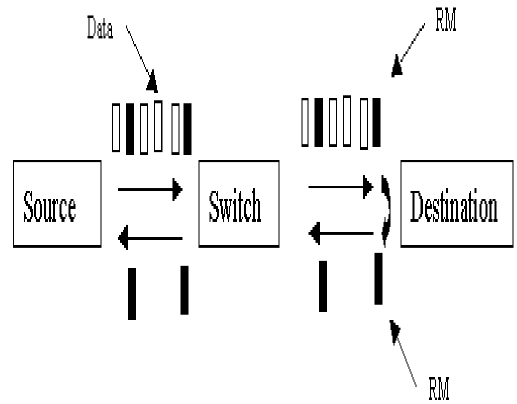


Figure 2.4: ABR Traffic Management Model

bandwidth is not available. The bandwidth available from the network may vary but shall not become less than MCR. The source end system (SES) is allowed to send data at the Allowed Cell Rate (ACR) which is less than the negotiated PCR. Immediately after establishing a connection ACR is set to an Initial Cell Rate (ICR), which is also negotiated with the network. The source sends an RM cell after transmitting every Nrm-1 cells (default Nrm value is 32). The RM cell contains a current cell rate (CCR) field initialized with the current ACR, an ER field, a CI bit, NI bit and few other fields which are not relevant to the discussion here. Feedback is provided to the SES that indicates at what rate a source is allowed to transmit. The switches in the network send this feedback by modifying the fields in the RM cells. There are two modes of feedback possible here: **binary mode** and **explicit rate mode**. In **binary mode**, a switch indicates whether it is experiencing (or about to experience) congestion or not by setting the CI or the NI bit. With **explicit rate mode**, the switches signal explicitly by setting the ER field in the RM cell at what rate the source is allowed to send; this rate is usually calculated so as to obtain optimal utilization of network resources and fairness between sources. The RM cells return to the sources carrying the minimum value of ER set by all switches on the path. The ER mode involves more complexity in the switches but offers better performance and avoids the oscillatory behavior that the binary mode is prone to. Both schemes require an algorithm running in the switch to decide what feedback should be sent to the source. The switch algorithm and its parameters are chosen freely by the switch vendors, and are not standardized by the ATM forum. In both the modes, a variety of parameters determine the exact behavior of the ABR algorithm making performance modeling and analysis difficult. When the source receives the RM cell from the network, it adjusts its ACR in accordance with the feedback received (whether in the form of ER or CI/NI). The ATM forum has defined a fairly complex set of rules as to how a source should respond to the feedback. When there is steady flow of RM cells in the forward and the reverse directions, there is a steady flow of feedback from the network. In this state, the control loop has been established and the source rates are primarily controlled by the network feedback (**closed-loop control**). When the source transmits data after an idle period, there is no reliable feedback from the network until after one round trip time. Hence for one round trip time, the source rates are primarily controlled by the ABR Source rules. Once the control loop is established the open-loop is replaced by the closed loop control. Since bursty traffic consists of busy and idle periods, open-loop control may be exercised at the beginning of every burst. Hence source rules assume considerable importance in ABR flow control. For a detailed description of the source, switch and the destination behaviors for ABR Flow Control, the readers are referred to [10]

It should be noted that both ABR data traffic and the available bandwidth for ABR are variable. Even though the average bandwidth is enough for an ABR connection in the long run, if there is not enough buffer to buffer the bursty traffic (either from VBR, which reduces the available bandwidth, or ABR itself, which requires more bandwidth), too many losses will result in low performance. For TCP over ABR, the switch buffer requirements and the proper tuning of the ABR algorithm and its parameters assume importance in order to maintain low cell loss rate and high performance.

2.2.2 Unspecified Bit Rate (UBR)

UBR service is designed for those data applications that want to use any left-over capacity and are not sensitive to cell loss or delay. Such connections are not rejected on the basis of bandwidth shortage and not policed for their behavior. Congestion control for UBR may be performed at a higher layer on an end-to-end basis. The switches almost do nothing for congestion control. They simply discard cells or packets whenever there is overflow.

The UBR service provides no specified bit rate, no traffic parameters and no quality-of-service guarantees. It offers only "best effort" delivery with no guarantees regarding cell loss, cell delay or cell delay variation. Devised originally to make use of excess bandwidth, UBR offers partial but inadequate, solution for those unpredictable bursty applications that don't readily conform to traffic contract parameters.

UBR's biggest deficiencies are its lack of flow control and inability to take other traffic types into account. When the network becomes overloaded, UBR connections go right on transmitting. Network switches can buffer some of this incoming traffic, but once buffers are full, cells are dropped. And because UBR connections have no traffic management contract with the network, theirs are the first cells to be dropped. UBR cell loss can be so high that "goodput" can easily fall to below 50%, an unacceptable level.

The UBR service may be enhanced by using intelligent drop policies in the switches. For TCP over UBR, these cell dropping policies and the switch buffer requirements are very important factors that influence performance.

2.3 TCP over ATM

In internetworking TCP/IP with ATM, network layer issues need to be distinguished from transport layer issues. Address resolution, routing and multicast support are the great challenges of the network layer that form an important research area. The transport layer issues concern with the interoperabilty of TCP/IP's transport protocols *viz* TCP and UDP, with ATMs transport layers. The issue here is mainly the traffic management and congestion control. Hence, in the discussion here, we are only concerned with the transport layer and congestion issues.

TCP over congested ATM network has arbitrarily low throughput performance [6, 16]. The dynamics causing this poor congestion behavior are explained below. Some of the throughput issues are rooted in the TCP protocol itself while others are due to the transport of TCP over ATM. In addition, the host characteristic such as the CPU and memory speed, operating system, system parameters and load often impose a limiting factor on the throughput.

TCP Packet Segmentation

One of the implications of carrying TCP traffic over an ATM network is that the ATM network will segment a packet into several smaller cells. Since the decision to drop data units when buffers fill up in switches and multiplexers is done on a cell by cell basis, fragments of packets can get dropped. This leads to the following two problems.

- **Fragmentation Problem**: When only a few cells from a packet are dropped, then the remaining cells of that packet form a fragmented packet. This fragmented packet will continue to be transmitted though the network towards the destination. Since the entire packet will be retransmitted anyway by TCP, the transmission of the fragmented packet constitutes a waste of bandwidth. The fragmentation problem is made worse by any factor that increases the number of cells dropped at the switch, such as small buffers, large TCP packets, increased TCP window size, or an increase in the number of active connections. Nevertheless, it should not suggest that the problem can be completely solved by appropriate settings of these parameters. Large buffers can result in unacceptably long delay; the beneficial effect of small windows, small packets or large buffers can be offset if the number of contending connections increases.

- **Scattered Loss Problem**: Consider the situation in which multiple cells are discarded from an overflowing buffer, and these cells belong to several different packets. This is essentially equivalent to having dropped all of the relevant packets since they will all need to be retransmitted. If the number of cells dropped is less than one packet size, then clearly it is preferable to drop one packet rather than many. This is called the scattered loss problem since the cell losses are scattered across many packets.

Synchronization effect

TCP window adaptation mechanisms are affected by increase in cell loss (and hence packet loss) due to ATM fragmentation of packets into cells. In many cases, multiple cells from different segments are dropped at one time due to congestion (the scattered loss problem mentioned above). When a packet loss occurs, the TCP slow-start and exponential back-off algorithm [14] are invoked which cause the TCP window size to decrease to a small number and the retransmission timeout to increase to a large number respectively. When multiple cells are discarded at the ATM switch, since cells from different packets are usually interleaved at the switch, these could more likely be from different packets belonging to different TCP connections using the network at the same time. Several TCP connections could then become *synchronized*, go through slow start at roughly the same time and wait to retransmit at roughly the same time. This can cause the network performance to oscillate. It can become congested when all the connections retransmit and become idle again when none of them transmit (all waiting for a retransmission timeout). This will result in a significant degradation in the TCP throughput.

Retransmission Timer Granularity

TCP/IP's control mechanism does not have the necessary granularity to perform well when losses occur at the ATM cell level due to switch buffer overflow. TCP traffic is very bursty. With small switch buffers, much less than the TCP window size, TCP sends small bursts of data separated by pauses [17]. These pauses are due to retransmission time-outs caused when TCP's window exceeds the switch's buffer space and the switch drops cells (and hence packets). Since TCP's minimum timeout is 500ms (few implementations have a 200ms granularity [12]), TCP spends far more time waiting to retransmit than it does sending data. This inactivity period of TCP results in significant underutilization of the link. Thus in a high-speed low propagation delay environments such as the ATM LANs, a large TCP granularity contributes to poor TCP performance. On the contrary if the granularity is too small, then unnecessary retransmissions will occur again degrading the TCP throughput. However, some TCP implementations [12] are relying less on the timer (e.g. Reno TCP), so the situation is improved to some extent.

While the factors discussed above explain why TCP performance over an uncontrolled ATM network is poor, in general, the performance is a function of a number of factors which can be broadly put as being **protocol considerations, host considerations and network considerations**.

Protocol Considerations: The specific implementation of the protocol and the protocol overhead places an upper bound on the throughput performance. Such factors include the inefficiency due to MTU (Maximum Transfer Unit) size and the TCP window management. In most cases, however, these limiting factors can be avoided with careful tuning and occasional modifications to the system parameters. Large MTUs are a performance disadvantage, due to increased number of wasted cells that the congested link transmits when the switch drops a single cell from one packet, though the processing overhead at end-nodes is less with large packets. The use of small MTUs minimizes the number of cells to be retransmitted in the case of packet discard, but it limits the throughput due to excessive overhead and increased packet processing. The typical maximum window size of 64KB or less, severely limits throughput for long ATM connections. Nevertheless, it is possible to extend the maximum allowed TCP window size, as it is provided as system option by some operating systems.

Host Considerations: The host characteristics such as the processing speed (CPU), memory speed, operating system, system parameters and load, buffering capabilities at the ATM interface often make the host as the limiting factor when transmitting or receiving over a high-speed network.

Network Considerations: Factors such as propagation delay, switch buffer sizes, link speeds, bandwidth mismatches, and congestion control methods in the network will determine the throughput that can be expected.

Our study will focus only on the network consideration issues specifically the potential capabilities of the ATM layer congestion control mechanisms for supporting TCP traffic and we hence set the other parameters to their optimal values so that they do not form a bottleneck in introducing any kind of uncertainty for investigating our objective.

2.3.1 TCP over UBR

When TCP is used over UBR, there is only one control mechanism that is active: the TCP control. UBR provides no control other than responding to congestion by dropping cells. As was mentioned before UBR provides no guarantees as related to cell loss or delay. Hence the performance of TCP over UBR depends greatly on the dropping policies used in the switches and the switch buffer sizes. With UBR, the performance is very poor due to the effect of the fragmentation loss. UBR+EPD improves the throughput performance, but can result in significant unfairness. Early Packet Discard (EPD) [6] reduces packet loss in congested buffers and the amount of resource wastage from cells belonging to corrupted packets (packet which has lost at least one cell). EPD works by accepting cells into the buffer if either the buffer occupancy is less than a certain threshold (called EPD threshold) or if other cells belonging to the same packet have already been in the buffer. Several other dropping policies have been proposed and a number of enhancements to EPD have also been suggested, however they do not form a subject of our study, hence details of these schemes are not provided here.

The only degree of freedom to control traffic in UBR is through buffer allocation (which includes cell drop policies). ABR has additional degrees of freedom in terms of switch schemes and source parameters.

2.3.2 TCP over ABR

The assessment of TCP performance over ABR is even more complex than the TCP-UBR case and it is difficult to make general conclusions. We will highlight here some of the design issues involved. Since both TCP and ABR have their own mechanisms for flow control (in terms of TCP) or congestion control (in terms of ATM), running TCP over ABR is normally controlled by both sets of mechanisms. In TCP, the maximum traffic is controlled by the congestion window (CWND), while in ABR, the traffic is controlled by Minimum Cell Rate (MCR), Peak Cell Rate(PCR) and Allowed Cell Rate(ACR). A key factor that impacts user and network performance for TCP over ABR is how well the traffic management mechanism used in TCP end system and ATM end system and switch mesh together in providing good end-to-end performance.

ABR Effects on TCP flows

When TCP operates over the ABR service, there are two control algorithms active - the TCP window based control running on top of the ABR control [19]. Data which uses TCP is controlled first by the TCP "slow start" procedure before it appears as traffic to the ATM layer. Suppose there is a large file transfer running on top of TCP. When the file transfer begins, TCP sets its congestion window to one. The congestion window increases exponentially with time. Specifically, the window increases by one for every ACK received and over a round trip time, the congestion window doubles its size.

As shown in Figure 2.5, at the ATM layer, the TCP traffic is considered bursty. 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 ACK). The length of an active period doubles every round trip time and the idle period reduces considerably. Finally the active period occupies the entire round-trip time and there is no idle period. After this point, the TCP traffic appears as a continuous traffic stream at the ATM layer.

When sufficient traffic load is not experienced in the ATM switches, the switch ABR 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 bottle-necked by the TCP congestion window size and not by the ABR source rate. In this state, the TCP sources can be thought of as being **window limited.**

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

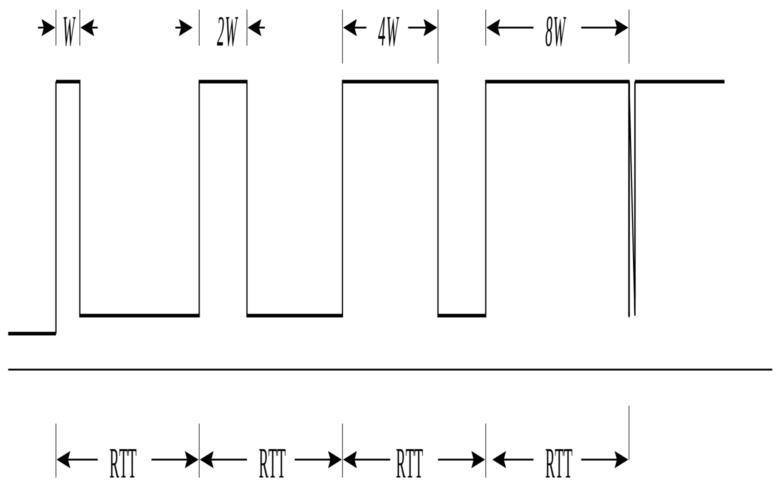


Figure 2.5: Nature of TCP traffic at the ATM layer

feedback asking the sources to reduce their rates. The TCP congestion window is now large and increasing. Hence it will send data at a rate greater than the source's allowed sending rate. The file transfer data is bottle-necked by the ABR source rate and not by the TCP congestion window size. In this state, the TCP sources are **rate-limited.**

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 fair-share values as computed by the ABR algorithm. 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. TCP achieves maximum throughput over ABR when there is no cell loss.

2.4 UDP over ATM

UDP is an unreliable transport protocol, typically used by real-time voice and video applications on the Internet and some applications like the Network File System(NFS) and Simple Network Management Protocol(SNMP). When UDP is used over ATM, the same fragmentation problem as in the case of TCP occurs, though the other effects discussed under TCP do not occur. Unlike TCP, UDP does not have any flow or congestion control features. When UDP is used over ABR, clearly an improved performance is expected because of the rate control of the ABR mechanism. UDP over UBR+EPD, on the other hand can give slightly better performance compared to UDP over ATM without any congestion control features.

Chapter 3

Experimental Framework

As was mentioned in Chapter 1, the goal of the study has been to make a comparison of the ABR and UBR+EPD services for TCP and UDP applications. In order to be able to make a fair comparison, the primary focus of the work was to design the range of experiments to clearly identify the conditions under which ABR or UBR+EPD would outperform the other. This Chapter discusses the design of the experimental scenarios, the rationale for the selection, the issues involved and the performance measures.

3.1 Experimental Scenarios

In order to do the performance evaluation, two sets of network environments have been chosen:

- End-to-End ATM networks i.e the network is homogeneous.
- Non end-to-end ATM networks i.e networks in which TCP/IP clouds are interconnected by an ATM cloud.

Figures 3.1 and 3.2 show the networks of the experimental scenarios. In an end-to-end ATM network, the TCP source and the destination are directly connected to the ATM network. In such a scenario, if ABR service is used, the source (the ATM Network Interface Card) can only send out data at the rate indicated by the network i.e., at the rate limit imposed by the ABR scheme. This can be thought of as a direct control of

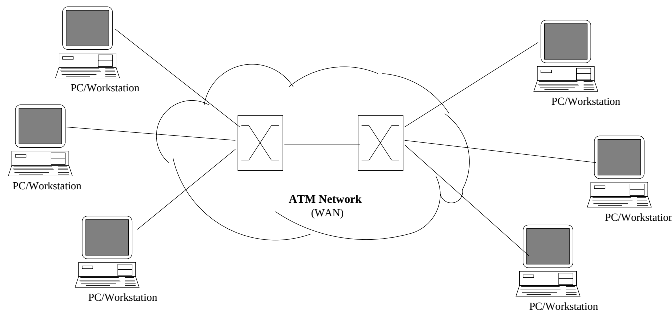


Figure 3.1: TCP/IP over an end-to-end ATM network

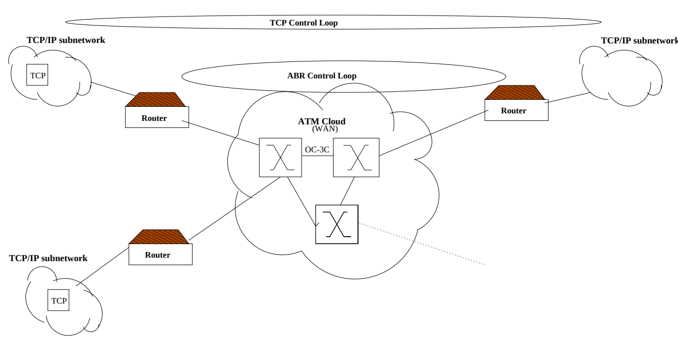


Figure 3.2: TCP/IP subnetworks interconnected with ATM backbone

TCP sources by the ABR scheme. Any data that is sent by the TCP source in excess of the allowed rate is queued in the host's send buffer and is not queued in the ABR end system buffers. When there is no more space in the send buffer the sending TCP process is put to sleep in the sending host because TCP does a blocking send and the data will go out of the TCP host's local disk to the ATM ABR NIC buffer only when there is sufficient space in the buffers. In such cases, the network interface cards or the ABR end systems need not have large buffers. Also, since the ABR control algorithms are usually well-designed to maintain small queues in the switches, the switch buffers also need not be large. Since the ABR mechanism was designed to share the available bandwidth fairly and efficiently among the connections, fairness and efficiency would be guaranteed. Also, as cell loss is minimized, the effect of fragmentation becomes negligible. In essence, in such cases, it is expected that ABR should outperform UBR+EPD. Our first set of experiments have been designed to verify this hypothesis.

While the end-to-end ATM network model allows us to test and compare the two services ABR and UBR+EPD in a simple way, these can only be seen as partially realistic. As the introduction of ATM will not cause other existing networks to just disappear, a more realistic model needs to consider the emergence of ATM provision in only some parts of TCP/IP based networks like the Internet. The second set of our experiments were designed to evaluate the performance of TCP in an internetwork where the end systems are connected to legacy LANs such as Ethernet or Token Ring, with the LANs interconnected through a wide area ATM network. In such a model, ATM can be considered to be used only as a backbone where an ATM cloud interconnects the various TCP/IP subnetworks (Figure 3.2). Supporting end-to-end Quality of Service and Traffic Management become important issues in these cases. In such internetwork environments, the ATM control mechanisms *viz.,* the ABR and UBR+EPD are applicable to the ATM subnetwork while the TCP flow control extends from end-to-end. If ABR service is used in the ATM subnetwork, then the edge routers' ATM network interface card (NIC) connecting the TCP/IP sources with the ATM cloud become the virtual ABR sources and control the transmission rate into the ATM subnetwork. The edge routers NIC may not be able to directly flow control the TCP sources except by dropping cells. In such cases, it is expected that ABR simply pushes congestion to the edge of the ATM network. The data sent by the IP source increases the NIC buffer occupancy and if the IP source is not notified of the NIC buffer's congestion, the buffer will overflow. The IP router at the edge of the ATM cloud acts as a barrier to the feedback from the network to the source, effectively defeating the purpose of the ABR mechanism by hiding the network congestion signals from reaching the end host. ABR expects the traffic sources to control its output rate in accordance with the feedback. But in this LAN interconnection or the non-end-to-end ATM model, though the edge routers ATM NIC will respond to the feedback and reduce its input rate into the ATM network, the IP source will not do so as the TCP sources feeding data into the IP router are not receiving this feedback and continue to send out data. The TCP source will only start to drop its rate after cell loss occurs in the edge router's ATM NIC, in turn resulting in a packet loss. Without special buffer management schemes the connection which finally has its packets dropped will not necessarily be the one whose rate was lowered by the ABR mechanism, leading to wrong connections decreasing its window size and output rate. This is particularly true when there are multiple TCP connections flowing through the same router. In summary, even if the ATM subnetwork is kept free of congestion by the ABR flow control, the end-to-end performance perceived by the application may not necessarily be better. Though the ABR rate control may allocate bandwidth fairly among the connections at the ATM level, the effective use of the allocated bandwidth by TCP is not guaranteed. This is basically a mismatch between the ABR mechanism which relies on the explicit rate feedback, and the TCP congestion control mechanism, which relies on the implicit feedback in the form of packet loss. This problem becomes more serious especially in the case of more stressful conditions such as a high-speed WAN. We may question the benefits of ABR in such cases and argue that UBR+EPD might be equally effective or even better and much less complex than ABR. Hence for a fair comparison between ABR and UBR+EPD, we have conducted experiments in the non end-to-end ATM network scenarios as well in order to verify the above analysis and hypothesis.

In each of the scenarios described above, we have further experimented with different network configurations for the ATM subnetwork. The details of the specific network configurations and the specific issues that each of these have been designed to address are provided in Chapter 4.

3.2 Issues Addressed

While the comparison of the two service classes could be done by varying a wide variety of parameters such as the TCP parameters, buffer sizes, feedback delays, round-trip times, algorithm parameters etc, and the parameter space is in fact very large, this work explored the dependency of the performance of ABR/UBR+EPD with UDP/TCP sources with respect to

1. Number of Connections or equivalently, load on the ATM network
2. Number of Congested links
3. Location of the Congestion Points.

3.3 Performance Measures

The performance metrics used in the study are defined below.

1. **Goodput**:

   $$Aggregate\ Goodput(Mbps) = Sum\ of\ goodputs\ of\ individual\ connections.$$

   The TCP goodput is measured at the destination TCP layer and is defined as the total number of good packets received at the destination application over certain duration of time. Goodput does not include packets that are part of retransmission or an incomplete packet. Also the the loss of throughput due to partially filled ATM cells is ignored in the goodput computations.

2. **Efficiency**:

   $$Efficiency(\%) = \frac{Aggregate\ Goodput}{Maximum\ Theoretical\ Goodput}$$

Where Maximum Theoretical Goodput is the maximum throughput measured at the TCP/UDP layer that can be achieved on an OC-3c link accounting for all the information overheads.

Accounting fot the information overheads the maximum theoretical goodput achievable with UBR on an OC-3c link is 134.513 Mbps while with ABR it is 130 Mbps, due to the RM cell overhead. In all the efficiency values reported in this study, these values are used for the denominator.

3. **Gain in Efficiency**: We define gain as the gain with ABR relative to UBR. Thus,

$$Gain = \frac{Efficiency\ with\ ABR}{Efficiency\ with\ UBR + EPD}$$

4. **Fairness Index**: For a service provider who provides quality-of-service to different classes of users at the same time, guaranteeing fairness is equally important as achieving a high effective goodput. With multiple users sharing a common link, it is very important to consider the fairness among the contending users.

   We have used Fairness Index as defined in [19] as measure of the fairness.

$$FI = \frac{(\sum x_i)^2}{n * \sum x_i{}^2}$$

   where x_i = Goodput of the ith TCP source
   n = number of TCP/UDP connections sharing the bottleneck link.
   A value of FI = 1 implies perfect fairness. FI is a normalized measure of the dispersion of the values of x_i.

5. **Link Utilization**: Measured for the bottleneck links and represents the amount of bandwidth on a link that is actually used. It is the cell-level link utilization and represents the link bandwidth used by both good and bad cells. It is reported as the percentage of the link bandwidth that is utilized.

6. **Cell Loss Ratio**: It is defined as

$$CLR = \frac{Number\ of\ cells\ lost}{Total\ number\ of\ cells\ transmitted\ during\ the\ simulation}$$

   Mean of the per-connection CLRs are reported.

7. **Packet Loss Ratio**: Packet loss ratios have been measured as the ratio of the number of packets lost to the number of packets sent out by the source during the simulation.

Chapter 4

Simulation Setup

Simulation models enable the performance evaluation of a system when mathematical methods are not available and experiments on the actual system are impossible or impractical. Evaluating the performance of TCP/IP over a high speed ATM WAN with ABR congestion control is not possible yet, since currently there are no ATM networks providing the ABR service. Also it is not feasible to develop mathematical models of such high speed networks. We therefore have carried out our investigation using simulation techniques.

This Chapter gives a description of the network models, configurations and the simulation parameters used in the study. All our simulation experiments are on a Wide Area Network (WAN). Experiments on a wide area network are especially important since the performance of TCP/IP over ATM is more sensitive in a wide area environment than on a local area environment. Also since LANs have shorter feedback delays and ABR control is quite effective in LANS, some properties of the ABR control mechanism may not be clearly observed in LAN configurations.

The simulation tool we used is the OPNET modeler tool [24]. OPNET allows the definition and modeling of a communication network in a hierarchical manner. At the highest level, the network topology and the connectivity are defined, along with several network parameters. At the next level, the protocols being used by each node as well as the way they communicate with each other are defined. Finally at the lowest level, the behavior of all the modules used in the network can be described using a state-machine representation. Each state in the state machine is described using C-language statements. The following sections describe the network configurations, traffic models, simulation parameters and the experiments performed.

4.1 Simulation Model

4.1.1 Network Models

An important element of a network protocol simulation is the choice of a suitable network topology. Different configurations highlight different aspects of the communication. The network models used in the simulation are described below. These network models constitute the ATM subnetwork in the homogeneous (end-to-end ATM) and the heterogeneous environments (non end-to-end ATM) mentioned in Chapter 3. For the heterogeneous environments, no specific technology is assumed for the legacy LANs, they are simply modeled as a single FIFO.

Two Node Configuration

Figure 4.1 illustrates the simulation model of a network with five connections. In this configuration there are two ATM switches connecting 5 TCP/UDP sources on one end to 5 TCP/UDP destinations on the other end (all of them pass through the same output port of the first switch). Each TCP/UDP host is connected to an ATM NIC component which performs ATM adaptation layer for data sources (AAL5) including segmentation and reassembly (SAR) of TCP/UDP packets. The ATM NIC component in addition has a cell buffer which performs the functions of a virtual ABR source and is equivalent to the ATM Forum's ABR source end system (SES). The service rate of this queue is determined by the ABR algorithm and so significant queues may develop when the available bit rate is low. For UBR service, no queueing occurs in this module since all traffic is sent out at peak cell rate (PCR) which is usually equal to the line rate. The two ATM switches perform cell switching between their input and output ports. The switch architecture used is a non-blocking output buffered switch. On the receiving side, cells are reassembled and passed to the TCP/UDP destinations.

This configuration has a single point of congestion and a key objective of this model is to remain fairly simple for simulation purposes. **Although the two node topology is rather simple it is useful in clearly understanding the performance dynamics and to evaluate the performance as a function of number of TCP/UDP connections**. We use this topology as the baseline as it helps to obtain a clear picture of the dynamics of the network.

Figure 4.2 illustrates the network model for the non-end-to-end ATM scenario when there are multiple flows through the router.

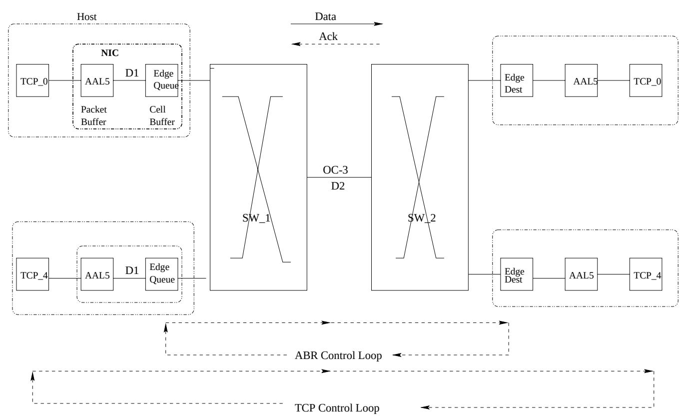


Figure 4.1: End-to-End ATM Scenario: Two Node Configuration

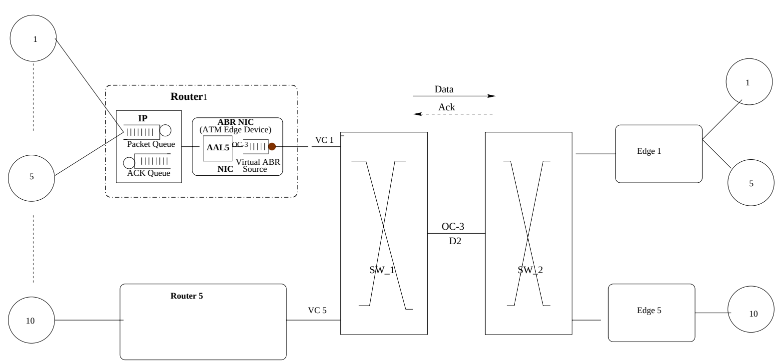


Figure 4.2: Non-End-to-End ATM Scenario: Two Node Configuration

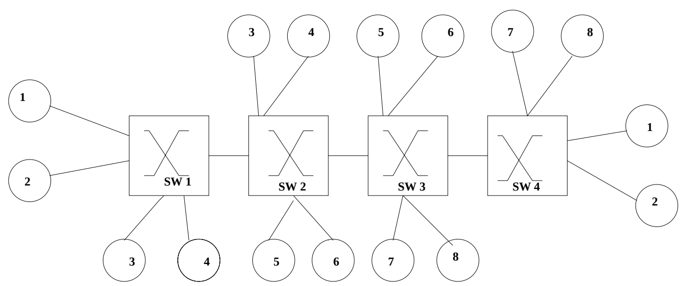
We have not modeled the detailed IP and SONET models as their impact is captured by simply accounting for their information overhead. We avoided the need for a SONET model at the physical layer by reducing the OC-3 link speed from 155.52 Mbps to 149.76

Mbps. A detailed IP model has also not been used because the IP routing functionality is not needed to evaluate performance of ATM networks. The IP MTU sizes are chosen in a such a way that no segmentation and reassembling is necessary at the IP layer. However in the simulations on heterogeneous networks (non end-to-end ATM), the IP router is modeled as a single FIFO queue that buffers the TCP packets. We also assumed that the physical links are perfectly reliable and error free so that packet/cell losses occur only due to congestion in the switches.

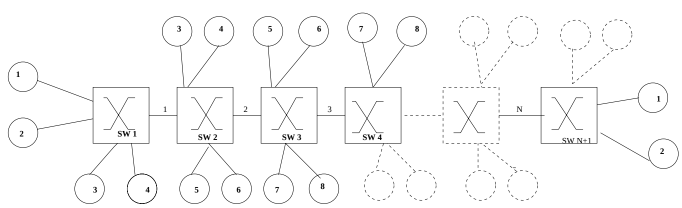
In the rest of the figures in this section, for the sake of simplicity, instead of illustrating the details of the components, the TCP and the ATM NIC components are grouped into a single module. Also when modeling the inter-networking environment where TCP/IP clouds are connected through the ATM cloud, the same topology is retained for the ATM network part. However, the TCP sources, would now feed traffic into the ATM network through a router instead of directly through the AAL5. The router could multiplex multiple TCP flows. The figures with the routers are not shown separately, as the topology is retained.

Multiple Node Configuration 1 (MNC1)

In real networks, if there is congestion it is likely to occur in multiple locations rather than in one single switch and the configuration shown in Figure 4.3 accommodates these situations. In this configuration there are four ATM switches each carrying four VCs. Both VCs 1 and 2 traverse multiple congested switches whereas each of the remaining VCs only traverse a single congested inter-switch link. Each link is traversed by 4 different VCs and a fair allocation would result in each VC utilizing 25% of the bandwidth. We use this simple configuration to compare the fairness characteristics of ABR and UBR+EPD for VCs passing through multiple congested switches. Note that the level of congestion in each of the switches is the same, as all the switches are carrying the same number of VCs. The circles represent a TCP or UDP source and we are interested in the throughputs of the VCs 1 and 2 that intermixes with cross-traffic at each ATM switch. All the links are of the same capacity. The round-trip delay in this configuration is 60ms. By varying the number of ATM switches along the path,



(a) Configuration with 3 links



(b) General Configuration with N links

Figure 4.3: Multiple Node Configuration 1

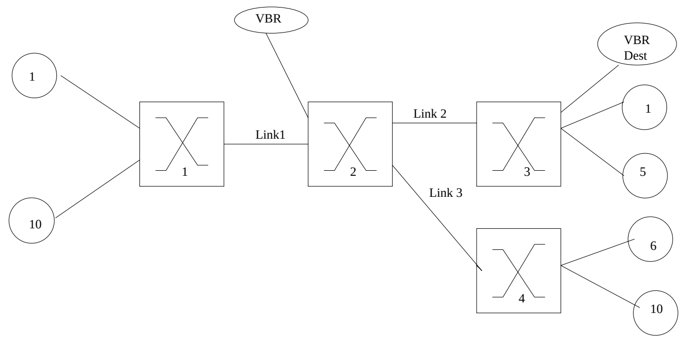


Figure 4.4: Multiple Node Configuration 2

we can **evaluate the performance as a function of the number of congested links/switches**. Figure 4.3(a) shows the configuration with 3 links and Figure 4.3(b) shows the configuration with 'N' links.

Multiple Node Configuration 2 (MNC2)

Figure 4.4 shows the second type of multiple node configuration we used in our study. This configuration consists of 4 switches and 10 Sources. Background VBR traffic flows through Link 2. This configuration is used to study the performance as a function of the location of the congestion point specifically when congestion occurs in a downstream node.

4.1.2 Traffic Models

1. **TCP Source**: The traffic model for TCP is a greedy (infinite) traffic source i.e., the source has always data to send. This model makes the application transparent, to better reveal the TCP behavior. However, due to TCP window constraint, the resulting traffic at the ATM layer will not be continuous and presents a variable load. Each TCP connection tries to send as much data as fast it can, i.e., all our TCP sources are persistent ones that generate packets at the link rate, as long as they are allowed to do so by the window mechanism, i.e., the sources can increase their sending rates up to their respective maximum transmission windows. The greedy traffic sources or the infinite traffic sources represents the worst case for

creating congestion, though they may not represent typical network traffic.

Here we have used the TCP-Reno version. The Reno version implements both the fast retransmit and the fast recovery algorithms.

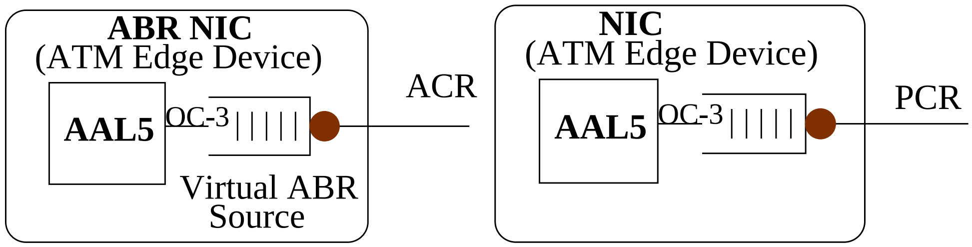
This model features the following TCP flow and congestion control features :

- Enhanced Retransmission Time Out(RTO) estimation based on both the mean and variance of the measured RTT [14]. A gain of 0.125 is used for the RTT estimators and a gain of 0.25 is used for the mean deviation estimators.
- Exponential Timer Backoff: Timeout value is doubled for each retransmission, with an upper limit of 64(This doubling is termed *exponential backoff* [2]).
- When a timeout and a retransmission occur, the RTT estimators are not updated when the acknowledgment for the retransmitted data finally arrives, i.e., A new RTO is not calculated until an acknowledgment is received for a segment that was not retransmitted (Karn's algorithm) [2].

It is assumed that the sender's congestion window and receiver's advertised window sizes are always identical. Also the simulation is based on packet entities rather than on the byte streams as in the actual TCP implementations. Such a packet based approach is simpler and is believed to be sufficient enough for investigating the dynamic behavior of the TCP and related flow and congestion control algorithmsi.

2. **UDP Source**: The UDP source has been modeled as a greedy source which generates packets at a constant rate. The packet size distribution and the inter-arrival times are chosen to be constant.

3. **Background VBR Traffic**: In order to observe the performance dynamics in the presence of non-ABR/non-UBR traffic, a background load is used in some of the experiments. This non-ABR/non-UBR traffic has priority over ABR/UBR traffic. The background VBR source is an ON-OFF source with exponentially distributed ON times and exponentially distributed OFF times (a two state Markov



(a) ABR Interface (b) UBR Interface

Figure 4.5: ATM Host Network Interface Card Model

Modulated Source). When ON the source transmits cells at PCR which is set as a fixed percentage of link rate and no cells are transmitted when OFF. In order to obtain the sufficient rate to fill an OC-3c link, and to achieve the effect of statistical multiplexing, a number of such sources are multiplexed together.

While this simple model may not be representative of real VBR traffic on networks, it does introduce a reasonable degree of variability in the usable ABR bandwidth.

4.1.3 ATM Host Interface (NIC) Models

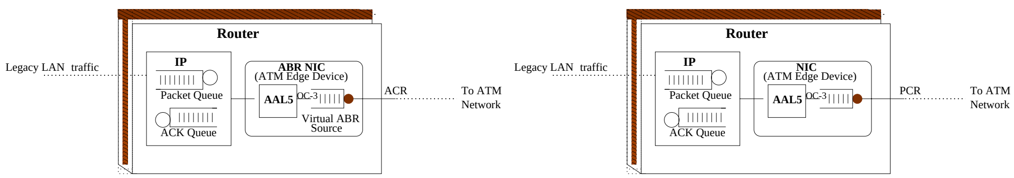
The host interface model (Figure 4.5) consists of two components - 1) AAL5 processor which performs ATM Adaptation Layer for data services *viz* segmentation and reassembly(SAR) of TCP packets. The SAR rate is set to the line rate. 2) a cell buffer (hereafter referred as the *Edge Controller*) which performs the functions of a virtual ABR source when ABR service is used for the ATM network and when UBR service is used, this cell buffer simply serves cells at peak cell rate of the UBR source (usually equal to the line rate, so that there is never any queue build).

Edge Controller: The edge controller connecting the TCP/IP sources with the ATM Network is modeled as virtual ABR source and the ATM cloud is connected to the TCP/IP destinations through virtual ABR destinations. This virtual ABR source assumes the behavior of an ABR source end system as specified in [10] and is modeled as a simple FIFO. That is, the virtual source sends a resource management (RM) cells every Nrm data cells and the segmented TCP packets that arrive at the virtual source are not simply passed on to the ATM cloud but are sent with the rate indicated in the backward RM cells (i.e., the service rate of the FIFO changes dynamically with the feedback rate received in the BRM cells). The backward RM cells that arrive at the virtual source are then removed from the connection. At the other end of the connection the virtual destination assumes the behavior of an ABR destination. Incoming RM cells are sent back to the originating virtual source and the data cells are passed over to the reassembly modules. There, the ATM cells are reassembled into TCP packets and are passed to the TCP/IP subnetwork. The behavior of the virtual ABR source is based on the use of ER scheme in the ATM network (It is assumed that the parameters CI and NI have no impact on the algorithm, it is a pure ER algorithm). Hence, there was no need to implement the binary mode rules of the SES, as binary mode ABR algorithms were not considered in this study. The model also features generation of out-of-rate RM cells when the allowed cell rate (ACR) is very low. However, in all our simulations, there was never a need for generating out-of-rate RM cells since ACR was never driven down to zero. Even when there was background VBR traffic, we always had some bandwidth left for ABR.

In the case of TCP over UBR simulations this Edge Router simply serves the purpose of a UBR End System and serves the cells at PCR of the UBR End System. If this PCR is equal to the line rate or the segmentation rate of the AAL5 modules then the source queues do not build up at the UBR End System, since all traffic is sent as soon as it arrives.

When modeling a host station connected directly to an ATM network (the homogeneous network environment), each TCP connection has its own edge controller and thus has its own ATM VC. When modeling the heterogeneous network environment (TCP/IP clouds interconnected by an ATM cloud), multiple TCP connections feed into a single ATM edge device (i.e the interface to the IP router) and will appear as a single VC at the ATM layer.

Figures 4.6 show the model of an IP router with ATM interface. This model is used in the simulation of heterogeneous networks (for TCP/IP over ATM backbone). The IP



(a) ABR Interface (b) UBR Interface

Figure 4.6: Model of an IP Router with ATM Network Interface

router is modeled as a FIFO queue which buffers the incoming TCP packets and serves them at line rate.

4.1.4 Switch Model

The ATM switches used in our network model are output buffered switches. Two stages of output buffering are supported, at the switch fabric and at the network interface modules. The switch fabric has four FIFO queues scheduled on a strict priority basis with CBR getting the highest priority, followed by VBR, ABR and UBR. Each output port supports per-vc queueing. The VBR VCs are given higher priority over ABR or UBR VCs. The individual VCs within a service class are served in a round-robin fashion.

4.2 Simulation Parameters

This section gives the system parameters used in the study and also the rationale for some of the parameter selections.

- **Topological Parameters**

  Link Capacity = 149.76Mbps (OC-3c after SONET overhead)

  Propagation Delay = 5 μs/Km.

  Distance from end-systems to switches = 10Km = 50μs delay.

  WAN1 : Switch-to-Switch Distance D2 = 1000 Km = 5 ms delay per link

[1]Round Trip time does not include the distance between the end systems and the switches which is assumed to be negligible

- **TCP Parameters**

  TCP type = Reno

  MSS = 9140 bytes

  Maximum Receiver Window size = 178KB = 20 segments (for an RTT of 10ms), 256 KB for an RTT of 60ms. (In experiments with higher round-trip delays, the window has been chosen large enough to fill the pipe)

  Timer granularity = 200 ms

  TCP Processing delay = 300 μs

  Initial RTO (max RTO) = 0.5 s

  Initial RTT (for the RTT estimator) = 0

  Initial Mean Deviation of RTT = 3 * Timer granularity

  Congestion Avoidance Mode = Linear

- **UDP Parameters**

  Segment Size = 9140 bytes

  Transmission Rate (unpaced) = 149.76 Mbps (= line rate)

  Packet Size Distribution = Constant

  Interarrival Distribution = Constant

- **IP Parameters**

  MTU size = 9180 bytes (default for Classical IP over ATM)

  Service rate = OC-3c (line rate)

- **AAL5 SAR Parameters**

  Segmentation Rate = 149.76 Mbps (= line rate)

- **UBR End System Parameters**

  PCR[2] = 149.76 Mbps

- **ABR End System Parameters**

  The ABR source end system parameters are shown in Table 4.1. In order to be

[2]In order to be able to exploit as much of the available ATM resources as possible we set the PCR to the link rate

able to exploit as much of the available ATM bandwidth as possible, we set the PCR to the link rate. The ICR is chosen to be a small value to avoid initial queue builds in the switches. MCR was chosen to be zero to avoid unnecessary reservation of bandwidth to the ABR connections. The rest of the parameters are the default values recommended by the ATM Forum [10].

- **Switch Parameters**
  - **Switch Core/Fabric Parameters**

    Core speed = 640 Mbps

    Core Memory Size = 4096 cells
  - **Output Port Parameters**

    Switch output buffer size (available to ABR or UBR)[3] = 8192 cells shared dynamically among the VCs on as needed basis.

    Reserved space for VBR = 1500 cells

    UBR-EPD threshold [4] = $\min\{0.8, \frac{switchi\ buffer\ size - 3*MTU}{switchi\ buffer\ size}\}$

4.3 Modeling Assumptions

- All traffic is unidirectional. Data transfer occurs only in the forward direction while reverse traffic consists of only the ACKs and the BRMs.

- There is no congestion in the backward direction. We assume that there is never any loss of feedback (no loss of BRMS). Since there is no data in the backward direction, the BRMs and the ACKs do not experience any queueing delay.

- The BRMs are not queued at all in the switches. They are forwarded as soon as they arrive into the switches.

- ABR and UBR services are not used simultaneously. That is we do not consider the cases where some TCP connections make use of the ABR service while others

[3]This is typical of ATM LAN switches today;for example, some provide a shared space of 8192 cells per netmod which is allocated to VCs of all traffic classes dynamically on an as needed basis.

[4]This choice of the threshold is based on the discussion in [?]

Table 4.1: ABR-End System Parameters

Parameter	Description	Value
ICR(neg)	Initial Cell Rate $ICR(used) = min\{ICR(neg), TBE/FRTT\}$	3 Mbps
PCR	Peak Cell Rate	149.76 Mbps
MCR	Minimum Cell Rate	0
ACR	Allowed Cell Rate	dynamically computed at the source
TCR	Tagged Cell Rate - upper bound on out-of-rate forward RM cells	10 cells/s
RDF	Rate Decrease Factor - fraction of PCR that is deducted from ACR upon arrival of a backward RM cell with CI = 1	1/16
RIF	Rate Increase Factor - fraction of ACR that is added to ACR upon arrival of a BRM cell with CI =0	1/16
ADTF	ACR Decrease Time Factor	0.5 s
CDF	Cutoff Decrease Factor	1/16
Nrm	Maximum number of data cells between two FRM cells	32
Mrm	Controls allocation of bandwidth between FRM-cells, BRM-cells and data cells.	2
Trm	Upper bound on the time between forward RM-cells	100ms
CRM	Missing RM cell count, Limit on the number of FRM cells which may be sent in the absence of received BRM-cells.	100 cells (TBE/Nrm)
TBE	Transient Buffer Exposure - determines the maximum number of cells that may suddenly appear at the switch during the first round trip before the closed loop phase of the control takes effect.	3200 cells (10% of Max shared buffer space)
FRTT	Fixed Round Trip Time	10ms for an RTT of 10ms

make use of the UBR service.

- The TCP delay ACK timer is not used. It is assumed that the segments are acked as soon as they are received at the destination layer.

Chapter 5

Results

The results of the TCP/UDP over ABR/UBR+EPD experiments are presented in this Chapter. First we present the validation results of the baseline configuration. Then we move onto the results of TCP/UDP performance over a network that is end-to-end ATM, for different network configurations. Finally, the results for the non-end-to-end ATM networks are presented.

5.1 Baseline Configuration

To determine the TCP and UDP performance in the absence of congestion, the baseline configuration was simulated. The baseline configuration consists of only one active TCP(UDP) session in Figure 4.1 of Chapter 4. The RTT used in these simulations was 10ms, i.e, the inter-switch link distance used was 1000 Km and the TCP window size used was 20 segments (approx 178 KB >= RTT * BW available to TCP connection).

Calculation of Maximum Theoretical Throughput

For the calculation of maximum theoretical throughput, only the overhead imposed by the PDUs format is taken into account. Other sources of overhead related to the operating systems analysis and inter-process communications are beyond the scope of this study.

With a MSS of 9140 bytes, the ATM layer receives 9140 bytes of data + 20 bytes of TCP header + 20 bytes of IP header + 8 bytes of LLC (Logic Link Control)/SNAP (Sun-Network Access Point) header + 8 bytes of AAL5 header = 9196 bytes. This will result in 9196/48 = 191.58 i.e., 192 cells at the ATM layer (with padding in the last cell). Thus each TCP segment results in 192 * 53 = 10176 bytes at the ATM layer. Hence the maximum possible throughput at TCP layer = 9140/10176 = 89.819% of the available bandwidth at the ATM layer. The OC-3c link capacity is 155.52 Mbps. The total length of the SONET OC-3c frame is 9*(9+261) = 2430 bytes and the total overhead consists of 90 bytes (27 bytes section overhead + 54 bytes of line overhead + 9 bytes of path overhead). Thus the available bandwidth at the ATM layer is $\frac{(2430-90)}{2430} * 155.52 = 149.76$ Mbps. Accounting for SONET overhead the maximum theoretical TCP throughput obtainable is 89.82% of 149.76 = 134.513 Mbps.

When ABR Service is used the maximum achievable throughput is further reduced because of the RM cell overhead. Since every 32nd cell was an RM cell, in our case it is 89.82% * 31/32= 87.01% of ABR capacity. In the absence of high priority background traffic (ABR capacity is fixed), this value is 87.01% of 149.76 = 130.306 Mbps. Thus the maximum throughput we expect to measure in the case of TCP/UDP over ABR is 130.306 Mbps.

The respective values are the same for UDP throughput over UBR+EPD and ABR (the UDP header is only 8 bytes though). For ease of comparison, the MSS for UDP is chosen to be the same as that for TCP.

Simulation Results - Goodput

The Goodput results for both UBR+EPD case and the ABR case for TCP and UDP and are shown in Table 5.1. The measured goodput does not include cells that are part of retransmission or an incomplete packet. The loss of throughput due to partially filled ATM cells is also ignored in the goodput computations. The goodput measurements were taken with a sampling interval of 0.05 seconds and the mean values over the entire duration of the simulation are reported here.

The simulation results closely match with the theoretical values obtained above, thus validating our models.

	Theoretical ($Mbps$)	Simulation ($Mbps$)	% error (Mbps)
TCP-UBR	134.513	133.8	0.5
TCP-ABR	130.306	129	1.004
UDP-UBR	134.513	134.513	0
UDP-ABR	130.306	130.3	0.095

Table 5.1: Goodput under Congestion-free Conditions (Baseline)

Buffering Requirements for Zero Loss

Further validation was done by running the experiments with 5 sources and 10 sources and with infinite buffers. This gave an insight into the switch buffer and NIC buffer requirements for zero loss and the observations were consistent with those seen in [18]. Table 5.2 contains the results for TCP running over UBR and ABR services with infinite buffering. All the connections achieved 100% of their fair-share of the maximum possible throughput and perfect fairness as there is zero loss.

It can be seen from Table 5.2 that in the case of UBR the queues build up in the switches while in the case of ABR the queues build up at the edge devices' ATM NIC buffer. The maximum queue size numbers give an indication of the buffer sizes required at the switch to achieve zero loss for TCP. In the case of UBR, for the 5 source configuration, the maximum queue size is 17200 cells. This is slightly less than the sum of the TCP window sizes[1](3840*5 = 19200 cells). This is because the switch has 1 RTT to clear out almost 2000 cells of TCP data before it receives the next window of data. The increase in buffer requirements is proportional to the number of sources. The maximum queue is reached just when the TCP connections reach the maximum window. After that, the window stabilizes and TCP's self clocking mechanism puts one segment into the network for each segment that leaves the network. In general for TCP over UBR, for zero loss in the switch, it can be seen that the amount of buffering

[2]The edge queue values are for each VC

	No. of Sources(N)	Queue Size at Switches (*cells*) B_s		Queue Size in Edge[2] (*cells* B_e)		B_s+N*B_e	Mean delay (ms)
		Max	Mean	Max	Mean		
TCP -UBR	5	17,200	13,320	0	0	17,200	65
	10	34,453	26500	0	0	34,453	100
TCP - ABR	5	4031	2745	2783	2270	17,937	64.8
	10	5022	2836	3183	2270	36,852	101

Table 5.2: TCP over UBR Vs ABR: Buffer Requirements for Zero Loss

required is approximately N * Maximum TCP window size, where N is the number of connections passing through the switch.

For ABR, the queue build up in the switches is well within the operating region of the ER algorithm (preset to 2880 to 6720 cells). However, at the edges, one receiver window's worth of buffering per VC is required to avoid losses. We see that the overall buffer requirements for ABR and UBR is the same (B_s + N*B_e is almost the same in both the cases, where B_s is the switch buffer size and B_e is the edge buffer size). 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. Also note that the edge buffer NIC requirements in the case of TCP over ABR holds true only in the case of a non-end-to-end ATM network environment. Such large buffers are not required in the case of TCP running over an end-to-end ATM network.

Also from Table 5.2, it can be observed that the switch buffer requirements for ABR is bounded and does not depend on the number of connections. This is quite intuitive because the ABR algorithm itself is designed to maintain the buffer occupancy within a preset middle region or the operating region. The operating region in this experiment was between 2880 and 6720 cells. It can be seen from Table 5.2 that the queue occupancy is stabilized around 3000 cells. It should be mentioned that the switch

[1] 1 Window = 20 MSSs of 9140 bytes each = 20*192 = 3840 cells

buffer requirements also depend on other factors *viz.,* 1) the round trip time 2) feedback delay 3) the transient response of the ABR algorithm and its parameter settings 4) nature of high priority traffic. We did not examine the effect of these parameters.

The mean delays can be seen to be approximately the same in the both the cases. This indicates that the queuing delays incurred in the edge device in the case of ABR is comparable to the queueing delay experienced at the switches in the case of UBR+EPD, hence the mean end-to-end delay is the same in both the cases.

So far we have considered the case where unlimited buffers were available and had established the buffer requirements for zero loss. However, in reality the buffers at both the edge device and the ATM switches are limited in size and much smaller than required for zero loss. Also since real switches carry 100's of VCs, and the number of VCs is dynamically varying, it is difficult to have buffering enough to avoid losses altogether for the worst cases of traffic scenarios considered here. Also large buffers means increased delay. The experiments to follow consider the effect of finite buffer sizes and are more realistic.

5.2 Results for Homogeneous Networks i.e., end-to-end ATM

It should be remembered that for all the experiments in this section, when the ABR service is used, the ABR loop goes all the way to the host containing the TCP source. It is assumed here that the ATM network interface cards have unlimited buffer space. This is a valid assumption for both UBR and ABR cases. For UBR case, there will not be any queueing in the NIC cell buffers as it is served at PCR which is set to the line rate and the incoming rate from the AAL5 is also at line rate. Some amount of queueing occurs in the NIC's cell buffer in the case of ABR, when the available bit rate is low. The TCP data stays in the send buffers itself when the ACR does not allow. So there will not be significant queue build in the ATM NIC.

5.2.1 Two Node Configuration - Performance as a function of Load on the ATM network

We begin our studies on the performance of the ABR and UBR+EPD schemes in the two node configuration illustrated in Figure 4.1 of Section 4.1.1 to understand the performance dynamics in the simplest network configuration and to study how the performance is affected as a function of the load or the number of connections. Each TCP connection has its own VC. With UBR service, each VC transmits data at the full line rate, while with ABR service, the rate is limited to the fair-share of the available bandwidth, as allocated by the ABR control algorithm.

The results for the case with 5 TCP sessions are summarized in Table 5.3 and for the case of 5 UDP sessions is summarized in Table 5.4.

	Aggregate Goodput ($Mbps$)	Efficiency (%)	FI[3]	CLR[4](%)	PLR[5](%)	Retrans %	LU[6] (%)
ABR	129	98.99	0.997	0	0	0	100
UBR+EPD	110	81.7	0.89	6.5	7.5	9.89	97.1

Table 5.3: End-to-End ATM Network Scenario - TCP over ABR Vs UBR+EPD: Two Node Configuration with 5 TCP sessions

	Aggregate Goodput ($Mbps$)	Efficiency (%)	FI	CLR(%)	PLR(%)	LU (%)
ABR	128.89	98.89	1	0	0	100
UBR+EPD	132.8	98.7	0.78	75.96	82.78	98

Table 5.4: End-to-End ATM Network Scenario - UDP over ABR Vs UBR+EPD: Two Node Configuration with 5 UDP sessions

[3] Fairness Index

[4] Cell Loss Ratio and the Packet Loss Ratio reported are the mean values over all the connections. These have been collected on a per-connection basis and include losses are various points in the network

[6] Mean Link Utilization

From Table 5.3 it can be seen that ABR clearly outperforms UBR+EPD both in terms of throughput or efficiency and in terms achieving perfect fairness for TCP traffic.

However, for UDP traffic, as can be seen from Table 5.4 the aggregate throughput was close to the theoretical maximum both with UBR+EPD and ABR. However, such high efficiency was achieved with UBR+EPD at the expense of marked unfairness. While few connections got high throughput, some got very low throughput as low as 2 Mbps. Both ABR and UBR+EPD can be seen to give the same performance if only the aggregate throughput is a concern. However, if fairness is desired then ABR clearly outperforms UBR+EPD.

The aggregate throughput achieved with ABR is close to the theoretical maximum for both UDP as well as TCP traffic, hence resulting in a high efficiency of 98.99%. This is because there was zero cell loss in the ATM network. Thus, ABR can be seen to be effective in achieving its goal of zero cell loss in the ATM network.

ABR service can be seen to provide nearly perfect fair allocation of bandwidth to each connection both in the case of TCP and UDP. The unfairness observed with the UBR+EPD can be attributed to the inherent nature of the EPD scheme itself. If the first cell of an incoming packet is discarded, all subsequent cells belonging to that packet are discarded in the EPD scheme. Moreover, the first cell of each incoming packet will be discarded until the buffer is reduced to below the EPD threshold. Some VC's packets will not be discarded once the EPD threshold of the FIFO queue is not exceeded and these VCs can be considered to be the "lucky VCs". To some degree the EPD scheme helps accelerate the data transmission of the "lucky VCs", while other TCP sessions detect packet loss and wait for the expiration of retransmission timer or duplicate ACKs. This causes unfairness in the bandwidth allocation among the VCs.

Effect of Loading or the Degree of Network Congestion

In order to determine how the performance gain seen with ABR in the experiment above scales with the load on the network, the number of connections in the two node model was varied from 5 to 25. The increase in the number of connections increases the network congestion. This section discusses the results of the effect of increased network congestion or the effect of load on the network on the scalability of the performance of ABR and UBR+EPD. The results from all these experiments are summarized in Tables 5.5 and 5.6 for TCP and UDP respectively.

	No. of Sources	Aggregate Goodput ($Mbps$)	Efficiency(%)	FI	CLR(%)	PLR(%)	Retrans. %
ABR	5	129	98.99	0.999	0	0	0
	10	128.91	98.95	0.998	0	0	0
	15	128.53	98.94	0.998	0	0	0
	20	129	98.99	0.997	0	0	0
UBR + EPD	5	110	81.7	0.89	6.5	7.5	9.89
	10	98.71	73.3	0.82	8.32	8.62	12.12
	15	90.21	66.9	0.76	9.6	10.2	16.78
	20	85.3	60.75	0.65	10.45	11.4	19.3

Table 5.5: End-to-End ATM Network Scenario - TCP over ABR Vs UBR+EPD: Effect of Number of Connections

Referring to Table 5.5, it is seen that with ABR, efficiency is unaffected by the increased load. There is near perfect fairness even at increased loads. Thus ABR mechanism can be seen to handle the increased loads very efficiently. With UBR+EPD, both efficiency and fairness are lower than in the case of ABR and degrade further with increased loads.

	No. of Sources	Aggregate Goodput ($Mbps$)	Efficiency(%)	FI	CLR(%)	PLR(%)
UBR + EPD	5	132.8	98.7	0.78	75.96	82.78
	10	131.07	97.4	0.4	77.8	83.45
	15	130.5	97.02	0.38	77.95	86.5
	20	130.24	96.82	0.27	78.5	87.6

Table 5.6: End-to-End ATM Network Scenario - UDP over ABR Vs UBR+EPD: Effect of Number of Connections

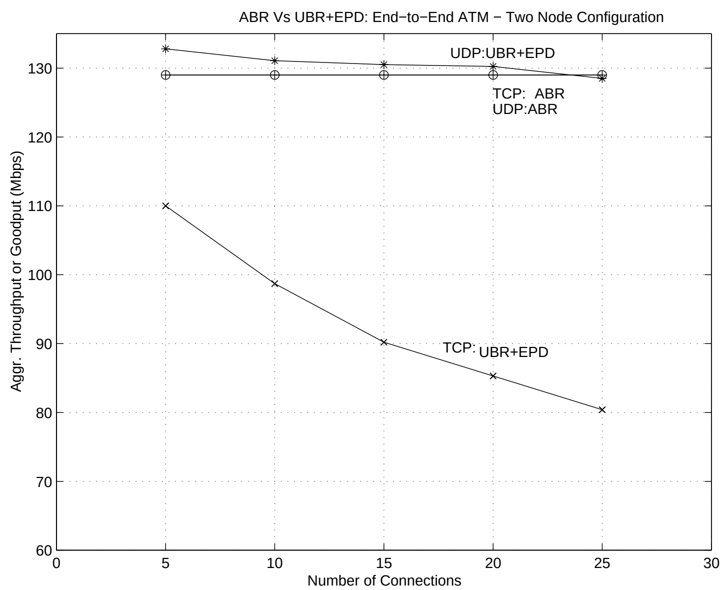
Referring to Table 5.6, for UDP traffic over UBR+EPD, though the efficiency or the aggregate throughput is not affected by the increased load, fairness shows worse values at higher loads. On the other hand with ABR, since there was zero loss in the switch due to ABR control and also in the edge because of infinite buffering as assumed for the end-to-end ATM cases, ABR efficiency was close to theoretical maximum and near perfect fairness, thus implying that the performance scales very well with increased load.

Figures 5.1(a) and 5.1(b) show the throughput and fairness index in the two-node model, as a function of the number of sources. Figures 5.2(a) and 5.2(b) show the efficiency and gain in efficiency in the two-node model, as a function of the number of sources.

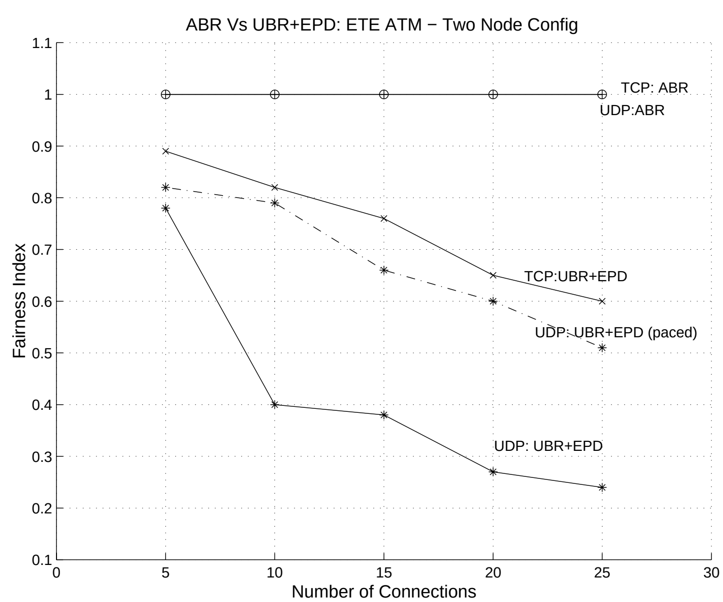
Referring to Figures 5.1(a) and 5.1(b), the throughput and fairness curves are nearly flat indicating that the increased load on the network has no effect on ABR performance. ABR can thus be seen to scale well with the load on the network. Similar observations can be made for the UDP over ABR too. The aggregate throughput deteriorates in TCP over UBR+EPD as the number of connections increases. Similar observations can be made with regard to UDP traffic, the aggregate goodput decreases, but the absolute values are much lower. In the case of UDP over UBR+EPD, though the aggregate throughput decreases with the increase in the number of connections, which can be attributed to the increased cell loss by the increase in the load on the network, the degree of deterioration is not as drastic as it was TCP over UBR+EPD. One possible explanation for this, in the case of TCP, it is the retransmissions that contribute more to the degradation in throughput than the cell loss itself.

With ABR, the fairness is unaffected by the increased load. The fairness index results re-establish the inherent unfairness of EPD both for TCP and UDP. With UDP fairness indices were much lower than in the case of TCP over UBR+EPD and fairness index can be seen to decrease with the increased congestion in the network.

Referring to Figure 5.2(b), there is always a gain in using ABR over UBR+EPD for TCP traffic and the gain is of higher magnitude at higher loads. Also the gain can be seen to increase more or less linearly with the load on the network. On the other hand,

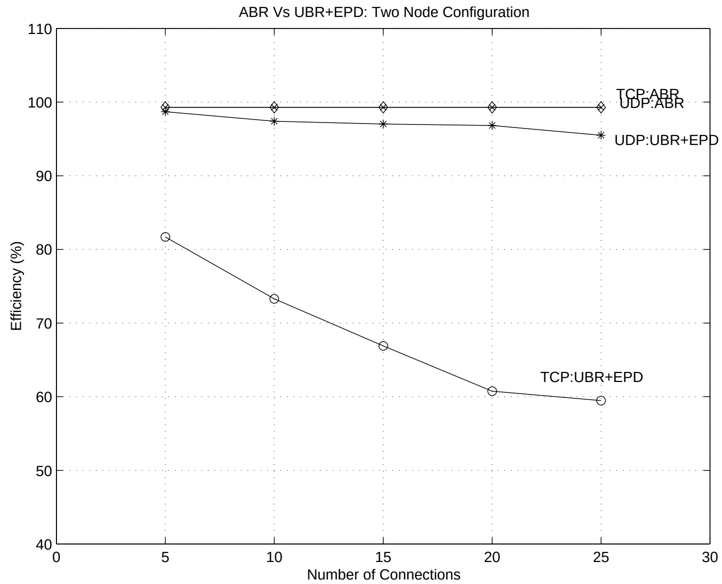


(a)

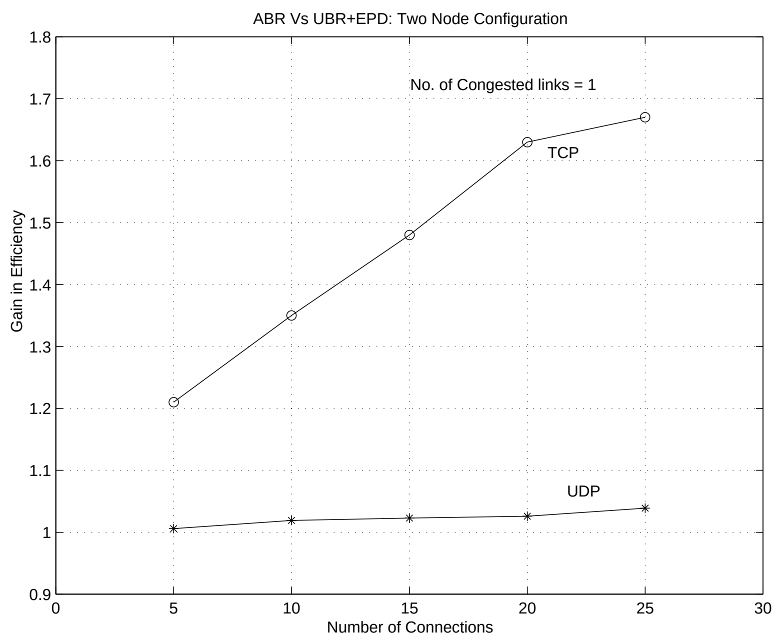


(b)

Figure 5.1: End-to-End Network Scenario: Effect of Number of Connections on Throughput and Fairness Index



(a)



(b)

Figure 5.2: End-to-End Network Scenario: Effect of Number of Connections on Efficiency and Gain

for UDP the gain in efficiency is just about 1 and does not seem to increase much by the increased network congestion. This implies that for this simple configuration with single congested node, both ABR and UBR+EPD give same performance in terms of efficiency for UDP traffic.

Comparing Tables 5.5 and 5.6, we see that in the case of UDP over UBR+EPD, the packet loss ratios were much higher than in the case of TCP. This reason being the open-loop nature of UDP over UBR+EPD. TCP sources on the other hand reduce their input rate upon detecting packet losses and hence in the steady state, the congestion state in the network is reduced. Further, though both the TCP and the UDP sources have been assumed to be greedy, the TCP sources offer a variable load to the network because of the implicit flow control mechanism, while the UDP sources always transmit at a constant rate offering the same load to the network. For the case shown in Table 5.6 it was assumed that all the UDP sources were transmitting at line rate. While this may not be realistic, it is the worst case for creating congestion.

Considering both the throughput and fairness, it can be concluded that the highest performance is achieved by using ABR over an end-to-end ATM network for TCP as well UDP.

Pacing UDP traffic

As was mentioned in the previous paragraph, TCP offers a variable load to the network, while the UDP load is constant. Hence in the discussion above, though the number of connections is the same both in the case of TCP and UDP, the load is not same. In order to compare the performance under similar load conditions, we paced the UDP source rates, so that we would see approximately the same amount of packet loss ratios as in the TCP case. While the pacing did not affect the efficiency, as efficiency was already close to theoretical maximum even in the worst case loading scenario for UDP, the fairness improved. Hence, only the fairness curve (Figure 5.1(b)) is shown for the paced UDP traffic. The fairness indices can be seen to be higher than in the unpaced case. Also for the end-to-end ATM scenarios, pacing affected only the UDP over UBR+EPD fairness performance. Packet level pacing does not have any impact for UDP over ABR, for the end-to-end ATM cases, as near perfect fairness and high efficiency were achieved even without pacing. This was because of the zero loss maintained by the ABR control in the ATM network and also since infinite buffering was assumed at the edge, there was no loss in the edge device. Hence only the fairness curve for the paced UDP over UBR+EPD is shown in Figure 5.1(b).

5.2.2 Two Node Configuration with Background VBR traffic

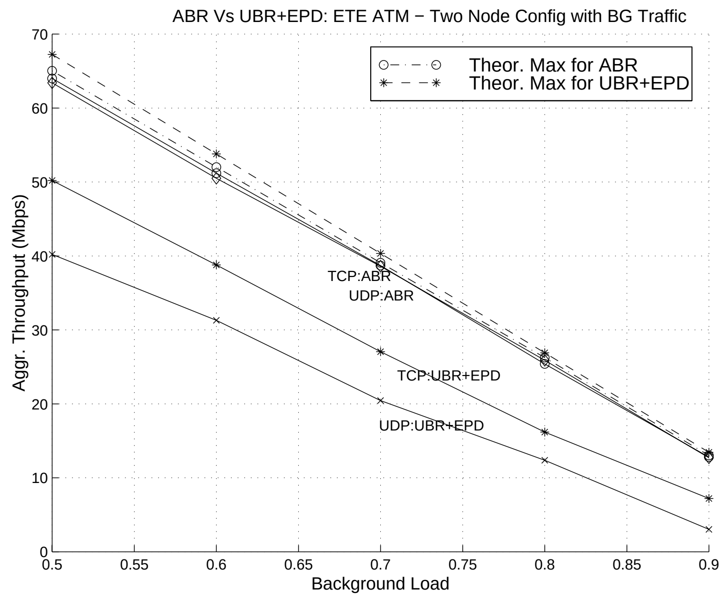
As ABR is intended to make use of the unused bandwidth in the network, it is evident that the presence of background traffic has great influence on ABR performance. The role of ABR in the network is to control the rate of ABR class traffic sources in the presence of non-ABR traffic so as to avoid cell loss in the network. To examine ABR's effectiveness at doing this, background traffic was fed on the bottleneck link in the two node configuration with 5 TCP/UDP sessions considered in the previous section. A description of the background traffic was given in Section 4.1.2. Ten such VBR sources were multiplexed to create sufficient variance in the available capacity. The performance obtained with ABR is compared with that obtained using UBR+EPD for various background loads.

Figure 5.3(a) shows the aggregate throughput obtained for a range of different background loads on the switch with ABR and UBR+EPD. Figures 5.4(a) and 5.4(b) shows the efficiency and gain in the same scenario.

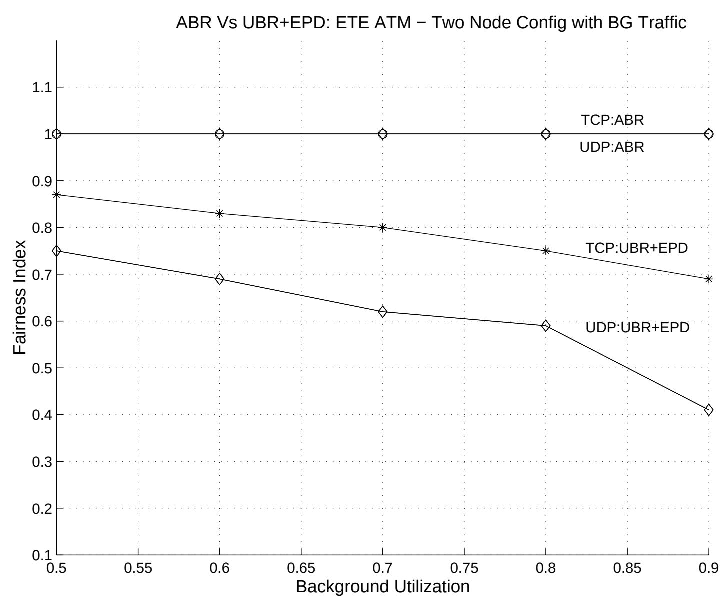
The results show that ABR consistently outperformed UBR even in environments with dynamically changing available bandwidth.

We see from Figure 5.3(a) and 5.3(a) that ABR achieves close to theoretical maximum throughput and gives better throughput and better efficiency than UBR+EPD both for TCP and UDP for various background loads. This is again due to the fact that ABR is successful in maintaining zero loss.

The efficiency curve for ABR is more or less flat, whereas for UBR efficiency can be seen to decrease with increasing background loads. With UBR+EPD, increasing the background load increases the loss in the switch, whereas ABR successfully achieves its goal of avoiding loss in the switch.

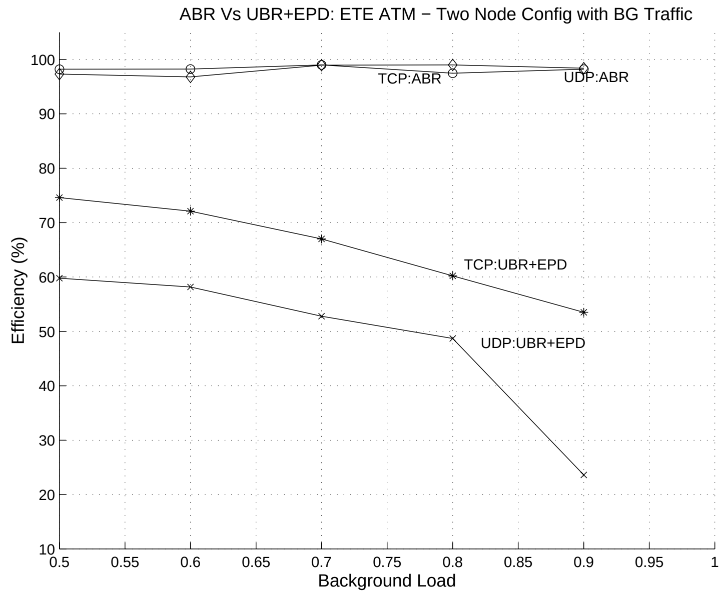


(a)

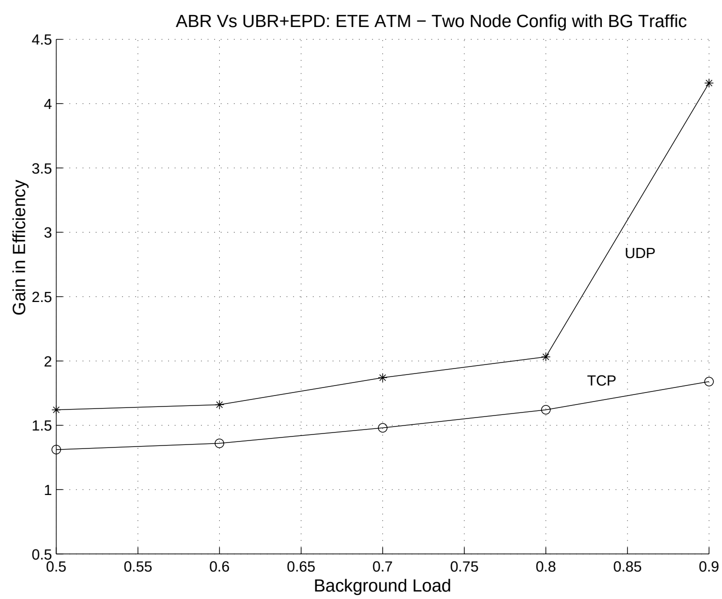


(b)

Figure 5.3: End-to-End Network Scenario: Effect of Varying Background Load. Number of TCP/UDP sessions = 5



(a)



(b)

Figure 5.4: End-to-End Network Scenario: Effect of Varying Background Load. Number of TCP/UDP sessions = 5

Figure 5.3(b) shows the fairness in bandwidth sharing among the five TCP connections as function of the background load. Fairness index is very close to one in the case of ABR and is unaffected by the load. With UBR+EPD fairness index decreases with the increase in the background load both for TCP and UDP, the values being lower in the case of UDP over UBR+EPD.

Figure 5.4(b) shows the gain as a function of the background load. The gain in using ABR to UBR+EPD improves with increased background load or effectively the load on the network.

5.2.3 Multiple Node Configuration 1 - Performance as a function of no. of Congested Links

In order to make a performance comparison of ABR and UBR+EPD in a more realistic network model, the Multiple Node Configuration 1 shown in Figure 4.3 is used in this part of the study. A detailed description of the simulation configuration was presented in Section 4.1.1. The results from the simulations on this configuration are summarized in Tables 5.7 and 5.8 for TCP traffic and Tables 5.9 and 5.10 for UDP traffic.

Sources 1 and 2 represent the transit traffic and show more or less the same behavior. All the other connections are local traffic and pass only through the output buffer of one switch. That is sources 3-to-8 exhibit the same behavior. This is evident from the throughput values shown for these connections in the tables below. The aggregate throughput shown in the tables below represents the aggregate of the connections passing through the same output port. The FI value reported is also for the set of 4 sources that share the same link. Since in this configuration each link is shared by 4 connections, the fair share for each connection would intuitively be one-fourth of the available bandwidth.

It is seen from Table 5.7 that in the case of TCP over ABR, an efficiency of 98.99% is reached with a fairness index close to 1. Such high efficiency and high fairness index was the result of the ABR mechanisms' effectiveness in maintaining zero loss in the ATM network. With TCP over UBR+EPD, connections 1 and 2 suffered higher packet loss ratios relative to the two local sources. This was because connections 1 and 2

	Aggregate Goodput ($Mbps$)	Efficiency (%)	FI	CLR(%)	PLR(%)	Retrans %
ABR	129	98.99	0.999	0	0	0
UBR+EPD	98.8	73.45	0.89	2.15	12.3	28.75

Table 5.7: End-to-End ATM Network Scenario - TCP over ABR Vs UBR+EPD: Multiple Node Configuration 1

	ABR				UBR+EPD			
Source No.	Goodput ($Mbps$)	CLR(%)	PLR(%)	Retrans. %	Goodput ($Mbps$)	CLR(%)	PLR(%)	Retrans. %
1	31.75	0	0	0	15	2.3	15.6	34
2	30.33	0	0	0	18	3.5	16.3	38
3	29.75	0	0	0	32.8	1.5	7.6	22
4	32.25	0	0	0	33	1.3	8.2	21
5	30.15	0	0	0	33.2	1.3	7.2	21
6	29.98	0	0	0	32.6	1.5	8.2	22
7	32.25	0	0	0	34	1.7	6.7	22
8	31.95	0	0	0	31.8	1.3	7.2	21

Table 5.8: End-to-End ATM Network Scenario: TCP over ABR Vs UBR+EPD: Multiple Node Configuration 1 Statistics of Individual Connections

passed through more number of congested links, so the cells belonging to their packets were more likely to be discarded at the switches than other connections. Thus, with UBR+EPD the efficiency was only 73.45 with a fairness index of 0.89 only. An important observation here is that as connections 1 and 2 pass through a number of congested links, their throughput was much lower than the fair-share. EPD is inherently unfair, this unfairness is further worsened if there are multiple congested links in the path between the source and the destination. Thus for TCP traffic, the gain in efficiency in using ABR over UBR is 98.99/73.4 = 1.35. The relative fairness index is 1/0.89 = 1.12

Similar observation can be seen even in the case of UDP traffic. In the case of UDP over UBR+EPD, however, higher packet loss ratios were observed compared to the TCP sources and hence low aggregate throughput. This was because the UDP sources, unlike the TCP sources, do not reduce their input rate into the network when a loss occurs. Thus for UDP traffic, the gain in efficiency in using ABR over UBR is G = 98.99/53.97 = 1.86. The relative fairness index is 1/0.8 = 1.25. Pacing the UDP traffic to see approximately the same amount of packet losses as in the case of TCP resulted in improving the performance of UDP over UBR+EPD, while it did not change the performance with ABR, as with ABR the efficiency was close to theoretical maximum even without pacing. Hence the relative gain in efficiency decreases when the UDP traffic is paced, while the fairness improves. This can be seen in Figures 5.8 and 5.7.

	Aggregate Goodput (*Mbps*)	Efficiency (%)	FI	CLR(%)	PLR(%)
ABR	128.32	98.987	0.998	0	0
UBR+EPD	72.6	53.97	0.80	14.4	75.12

Table 5.9: End-to-End ATM Network Scenario: UDP over ABR Vs UBR+EPD: Multiple Node Configuration 1

The key result is that UBR+EPD has bias against connections passing through multiple congested switches. So in a network configuration such as the above ABR gives significantly better performance than UBR.

	ABR			UBR+EPD		
Source No.	Goodput ($Mbps$)	CLR(%)	PLR(%)	Goodput ($Mbps$)	CLR(%)	PLR(%)
1	32.25	0	0	9.9	20.4	89.3
2	30.23	0	0	14.3	16.2	80.2
3	29.90	0	0	20.17	9.1	60.6
4	30.33	0	0	28.23	11.9	70.38
5	31.65	0	0	21.30	10.3	59.3
6	30.97	0	0	27.0	10.7	71.68
7	32.23	0	0	28.4	11.4	69.8
8	29.86	0	0	29.9	9.6	61.18

Table 5.10: End-to-End ATM Network Scenario: UDP over ABR Vs UBR+EPD: Multiple Node Configuration 1 Statistics of Individual Connections

Effect of Number of Congested Links on the Throughputs of VC1 and VC2

In the Multiple Node Configuration 1 considered above there are 3 congested links. We extended this topology to study the effect of the number of congested links traversed by VC1 and VC2 on their throughput performance and the fairness indices of the ABR and UBR+EPD mechanisms. This general configuration was shown in Figure 4.3(b) of Section 4.1.1. Since both VCs 1 and 2 exhibit the same behavior the plots below show only the throughputs of VC1.

Figures **??** and 5.6 shows the throughput and efficiency for VCs 1 and 2 as a function of the number of congested links. With ABR the throughput is unaffected by the number of congested links in the path while with UBR+EPD, the throughput can be seen to be degrading as the number of congested links increases.

Figure 5.7 plots the fairness index against the number of congested links. The fairness index is 1 irrespective of the number of links in the case of ABR while the fairness becomes worse as the number of congested links increases in the case of UBR+EPD. The FI values show much lower values in the case of UDP than in the case of TCP.

Figure 5.8 plots the gain in efficiency as a function of the number of congested links. It can be seen that for the end-to-end ATM scenario the gain is more or less increasing exponentially with the number of congested links. The gain values are higher for UDP relative to TCP.

Pacing UDP Traffic

Note that in the above study, again UDP load on each link was much higher than the TCP load, though the number of connections is the same in both the cases. To make the comparison fair, we repeated the UDP experiments, to see approximately the same amount of packet loss ratios as in the case of TCP, so the load is approximately the same. It was observed that the UDP over UBR+EPD performance improves with pacing. Hence the relative gain of ABR to UBR for paced UDP is lower than in the unpaced case. This is shown in Figure 5.8. Also pacing improved the fairness for UDP over UBR+EPD as can be seen from Figure 5.7

These results indicate that ABR gives significant performance gains both in terms of throughput and fairness relative to UBR+EPD even in a multi-hop scenario, the gain being higher in the case of UDP relative to TCP

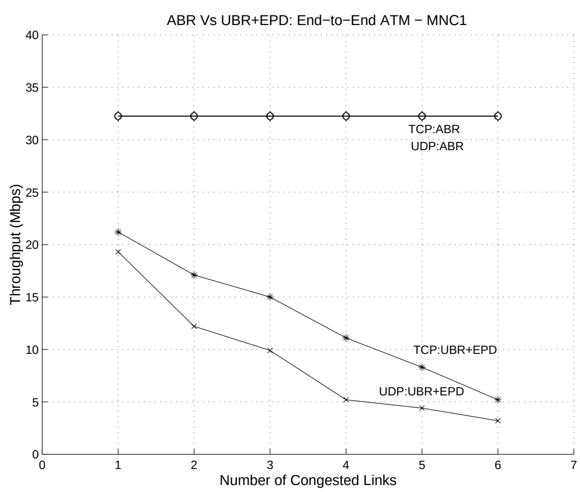


Figure 5.5: End-to-End ATM Scenario - Multiple Node Configuration 1: Throughput Vs Number of Congested Links.

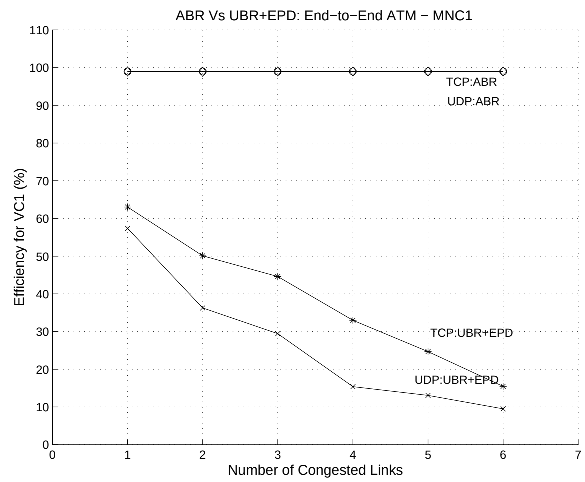


Figure 5.6: End-to-End ATM Scenario - Multiple Node Configuration 1: Efficiency Vs Number of Congested Links.

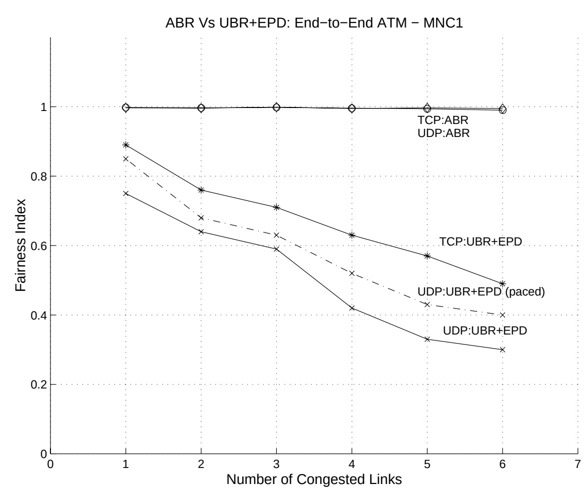


Figure 5.7: End-to-End ATM Scenario - Multiple Node Configuration 1: Fairness Index Vs Number of Congested Links.

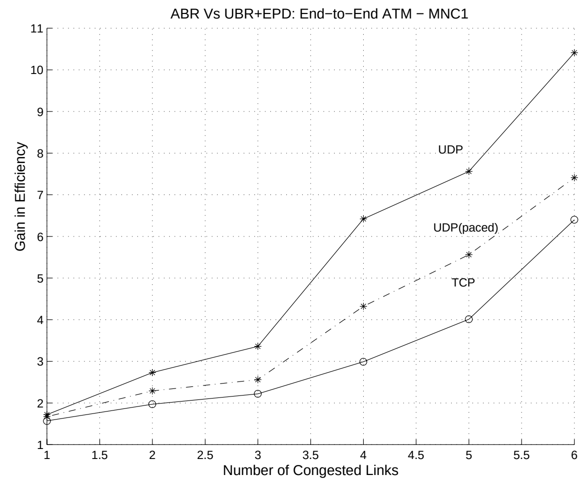


Figure 5.8: End-to-End ATM Scenario - Multiple Node Configuration 1: Gain Vs Number of Congested Links.

5.2.4 Multiple Node Configuration 2 - Performance as a function of the Location of the Congestion Point

Experiments were conducted on the Multiple Node Configuration 2 shown in Figure 4.4 to evaluate the ABR scheme and the UBR+EPD schemes with regard to the location of the point of congestion. Specifically we were interested in seeing the effectiveness of the two schemes ABR and UBR+EPD when congestion occurs in a downstream node. The results from the simulations on this configuration are presented in this section.

It is seen from the topology that sources 1-to-5 are bottlenecked at link 2, while sources 6-to-10 are bottlenecked at link 1. The fair share of the bandwidth for these connections is given in Table 5.11. Background VBR traffic uses up 100 Mbps of link 2's capacity leaving 50 Mbps for the ABR/UBR traffic.

Referring to Table 5.12, the aggregate throughput of all the 10 connections can be seen to be higher in the case of ABR than in the case of UBR+EPD.

With UBR+EPD, all the connections compete for the link bandwidth in link 1. However, many cells (and hence packets) of sources 1-to-5 that go through link 1 are discarded at switch 2 because of rate-mismatch at congested link 2. Hence the aggregate throughput of these connections was only 20.2 Mbps as opposed to an aggregate fairshare

Source	Fair-Share (Mbps)	Fair-share accounting for the overheads	
		ABR	UBR+EPD
1 to 5	50/5 = 10	8.7 each	8.98 each
6 to 10	(150-50)/5 = 20	17.4 each	17.96 each

Table 5.11: Fairshare of Bandwidths in Multiple Node Configuration 2

	Goodput T1-T5	Goodput T5-T10	Aggregate Goodput ($Mbps$)	Efficiency (%)	FI	Mean PLR	
						1-5	6-10
ABR	45.5	82.78	128.28	98.45	1	0	0
UBR+EPD	20.2	40.58	60.78	60.05	0.69	15.6	5.4

Table 5.12: End-to-End ATM Network Scenario - TCP over ABR Vs UBR+EPD: Two Node Configuration with 5 TCP sessions

	Link 1	Link 2
ABR	100%	100%
UBR+EPD	80%	95%

Table 5.13: End-to-End ATM Network Scenario - Multiple Node Configuration 2 : Mean Link Utilizations of the bottleneck links.

of 45Mbps. Also, sources 5-to-10, which should get an aggregate fair-share of 17.96*5 = 89.8 Mbps, if the algorithm were fair, received only 40.78 Mbps. This was because sources 1-to-5 took up the bandwidth at link 1, which ultimately was wasted due to the congestion at the downstream link 2. Link bandwidth in link 1 could otherwise have been used by connections 5-to-10. Hence the efficiency is very low with UBR+EPD. On the other hand, ABR flow control reduces the transmission rate at each of the edge device NICs for sources 1-to-5, so that these connections only occupy 1/3rd of the link bandwidth in link 1, allowing the other 2/3rd of this bandwidth to be used by the other connections.

The mean link utilizations are shown in Table 5.13 for both ABR and UBR+EPD cases. Link 1 is significantly underutilized in UBR+EPD case, while it is fully utilized in ABR case, because of the same explanation given above. Because of packet losses in the case of UBR+EPD, the TCP sources which eventually detect congestion reduce their window sizes and reduce their transmission rate into the network and this causes underutilization of link 1. On the other hand in the ABR case, since there are no losses at all, the TCP sources transmit at their maximum windows in steady state and both the bottleneck links 1 and 2 are fully utilized, with sources 5-to-10 contributing to 2/3 rd of the link utilization on link 1 and sources 1-to-5 contributing to 1/3rd.

Lack of fairness can be seen to be even more extreme for UBR+EPD in multiple-bottleneck scenarios with rate-mismatches.

The key result of this study is that when the point of congestion is located in a downstream node, ABR can outperform UBR because it can allocate different bandwidths to various connections according to their bottleneck rates while in UBR, the link bandwidth in upstream links may be wasted by packets that are discarded in a downstream congested link. It is clear that in large networks, where there are multiple bottlenecks and rate-mismatches, the benefits of ABR can be significant.

5.2.5 Summary of End-to-End Results

- In both single congested link and multiple congested link cases, ABR outperformed UBR+EPD.
- ABR scales well with increased network congestion.
- ABR scales well with increased number of congested links, in other words with the size of the network.
- In terms of fairness ABR always outperformed UBR+EPD.
- The gain in efficiency in using ABR over UBR is higher in the case of UDP relative to TCP when there are multiple congested links.
- ABR also scales well with respect to the location of the point of congestion, while UBR+EPD performs very poorly especially when the congestion point is located in a downstream node.

5.3 Results for Heterogeneous Networks

This section focuses on the results obtained on TCP/IP internetworks where the TCP end systems are connected to legacy networks, with the legacy networks interconnected through an ATM wide area network serving as a backbone. A general network scenario for such heterogeneous environments was shown in Figure 3.2. In this section, various topologies are considered for the ATM part of the internetwork. For ease of comparison, the internal topologies for the ATM cloud have been retained same as in the previous section. The total RTT of the connections in these environments is the delay in the local IP network plus the delay in the ATM cloud. Since we are assuming ATM WAN, the delay in the ATM segment dominates the RTT of the connection. All routers have single FIFO per output port. This study does not consider the per-flow queueing or intelligent scheduling in the router.

It should be remembered here that when ABR service is used, the ABR control terminates at the router's ATM NIC or the ATM access points. And due to limited buffering at the NIC, the NIC now forms the focal point of congestion. Note that any connection that goes through an ATM ABR network will experience only a single point of congestion. When UBR service is used, the point of congestion is still in the ATM switches in the network and hence each connection could see either a single congested switch or multiple congested switches depending on the network configuration. In order to make a fair comparison between the two service classes in such environments, the edge router's NIC buffer (B_s), which is the TCP/UDP over ABR's congestion point and the ATM switch buffers (B_s), which is the TCP/UDP over UBR+EPD's congestion point, have been kept to be same orders of magnitude ($N*B_e = B_s$, where N is the number of edge devices used in the set up, $B_s = 8192$ cells). We believe this is a fair test as the amount of buffer space available to UBR (in the switch) is equal to the amount of total buffer space available to ABR in the end systems NIC.

5.3.1 Two Node Configuration with single source connected to each Router

We first consider the two node configuration shown in Figure 4.1 with finite edge device NIC buffering. In this configuration each TCP source is connected to a unique edge device and there are 5 TCP sources resulting in a 1:1 loading on the edge device and 5:1 loading on the ATM switch. Though this is not realistic as in a real network a router carries multiple TCP flow each and not just one, this configuration is the simplest case to investigate the non-end-to-end ATM network scenarios. The results from this scenario are summarized in Table 5.14. Since the router carries only one flow, the UBR+EPD results are the same as in the end-to-end ATM case. It should be remembered that the losses occurred in the edge device in the case of ABR while in the case of UBR+EPD losses occurred in the switch.

For TCP traffic, referring to Table 5.14, both ABR and UBR+EPD achieve the same level of performance for TCP as far as throughput is concerned. In terms of fairness, again ABR achieved perfect fairness, while UBR+EPD was unfair. The fairness index was still very close to 1 in the case of ABR. This is attributed to the fact that in this experiment, each TCP session has its own edge device and hence the cell loss ratio

	Aggregate Goodput ($Mbps$)	Efficiency (%)	FI	CLR(%)	PLR(%)	Retrans %	LU (%)
ABR	100	76.7	0.999	5.6	12.72	10.69	64.4
UBR + EPD	110	81.7	0.89	6.5	7.5	9.89	97.10

Table 5.14: Non-End-to-End ATM Network Scenario - TCP over ABR Vs UBR+EPD: Two Node Configuration

	Aggregate Goodput ($Mbps$)	Efficiency (%)	FI	CLR(%)	PLR(%)	LU (%)
ABR	120.16	92.2	0.998	75.6	94.62	56
UBR+EPD	132.8	98.7	0.78	75.96	82.78	98

Table 5.15: Non-End-to-End ATM Network Scenario - UDP over ABR Vs UBR+EPD: Two Node Configuration

observed in the edge device for each of the TCP sessions and hence the packet loss ratio was the same, which in turn reflected as a fairness index close to 1, despite the losses. The bottleneck link utilization can be seen to be lower in the case of ABR than in the case of UBR+EPD. This is again due to the TCP over ABR sources experiencing the same amount of packet losses. The TCP over ABR sources behavior got synchronized leading to the link being occasionally idle when all of them were doing slow-start after detecting multiple packet losses. This contributed to the lower mean link utilization in the case of TCP over ABR in the non-end-to-end ATM scenario.

On the other hand, for UDP traffic, it can be seen from Table 5.15 that UBR+EPD outperformed ABR. The poor performance of ABR in this case is due to the fact that each of the five UDP connections suffered a PLR of 94.62% resulting in lower aggregate throughput. Such high packet loss ratios resulted from severe overloading on the edge. Also, due to lack of any intelligent packet dropping mechanism in the edge device NIC buffer, a single cell drop in the edge device NICs resulted in a very high percentage of corrupted packets which ultimately got discarded at the destination AAL5 re-assembly module and thus the PLR for each of the connections in the case of ABR was very high. The fairness index however was close to 1 and in fact in this scenario, it was the near perfect fairness in allocation of bandwidth that resulted in a low aggregate throughput. ABR does a fair allocation of bandwidth and hence in this scenario with 5 edge devices, each edge device gets a fair-share of about 30 Mbps, while the inflow into the edge is 150Mbps. Hence congestion occurs in the edge and cells are dropped.

So in a scenario such as the above UBR+EPD gave better throughput performance than ABR both for TCP as well as UDP traffic.

Effect of the Number of Connections

The same set of experiments as in Section 5.2.1 were repeated again, this time with finite buffering at the NIC buffer in the ATM interface of the router. Comparing Tables 5.5 and 5.16, it can be seen that UBR+EPD performance is the same. However, the performance with ABR can be seen to be degraded as compared to the non-end-to-end case, as there are now losses in the edge device due to limited buffering. Though the ABR mechanism is effective in fair allocation of the bandwidth among the various edge devices and in maintaining zero cell loss within the ATM network, the throughput performance seen at the application is less than the theoretical maximum because of the losses in the edge device. Further, since the edge device does not implement any kind of an intelligent packet discarding mechanism and simply drops cells, the fragmentation problem recurs and there is considerable wastage of bandwidth due to transporting packets that get corrupted because some cells have been lost in the edge. This is evident from the results shown in 5.16. It can be seen that average PLRs are much higher in the case of ABR than in the case of UBR+EPD even though CLRs are lower.

However, for UDP traffic, as can be seen from Table 5.15 UBR+EPD achieved better performance in terms of aggregate throughput, while fairness suffered.The performance gain for aggregate throughput was 63% relative to ABR. The reason for this is the inherent unfairness. Though the mean PLR can be seen to be as high as 82.3%, it was observed that few connections suffered very high packet loss ratios while others very

	No. of Sources	Aggregate Goodput ($Mbps$)	Efficiency(%)	FI	CLR(%)	PLR(%)	Retrans %
ABR	5	100	76.7	0.999	5.6	12.72	10.69
	10	94	72.1	0.997	6.3	18	22
	15	85	65.23	0.998	7.1	23.3	31
	20	76	58.32	0.999	8.3	25.3	38
UBR + EPD	5	110	81.7	0.89	6.5	7.5	9.89
	10	98.71	73.3	0.82	8.32	8.62	12.12
	15	90.21	66.9	0.76	9.6	10.2	16.78
	20	85.3	60.75	0.65	10.45	11.4	19.3

Table 5.16: Non-End-to-End ATM Network Scenario - TCP over ABR Vs UBR+EPD: Effect of Number of Connections

	No. of Sources	Aggregate Goodput ($Mbps$)	Efficiency(%)	FI	CLR(%)	PLR(%)
ABR	5	120.6	92.4	0.996	75.6	94.62
	10	120.2	91.2	0.997	84.5	95.6
	15	118.9	90.7	0.996	88.6	96.7
	20	117.99	90.4	0.996	89.7	97.3
UBR + EPD	5	132.8	98.7	0.78	75.96	82.78
	10	131.07	97.4	0.4	77.8	83.45
	15	130.5	97.02	0.38	77.95	86.5
	20	130.24	96.82	0.27	78.5	87.6

Table 5.17: Non-End-to-End ATM Network Scenario - UDP over ABR Vs UBR+EPD: Effect of Number of Connections

lucky and suffered very low packet loss ratios. Hence, while few connections achieved very low throughput others achieved high throughput. It is these lucky connections that contributed to the overall high aggregate throughput in the case of UBR+EPD. In the case of ABR, since each UDP session had its own edge device, the amount of congestion in all the five edge devices was the same, and hence all the five UDP sessions suffered approximately the same packet loss ratios. Further the mean packet loss ratio was much higher than in the case of UDP over ABR, again owing to the fact that all the UDP sessions suffered the same amount of cell loss in the edge devices and hence the same packet loss ratio.

Figures 5.9(a) and 5.9(b) show the throughput and fairness index in the two-node model, as a function of the number of sources. Figures 5.10(a) and 5.10(b) show the efficiency and gain in the two-node model, as a function of the number of sources.

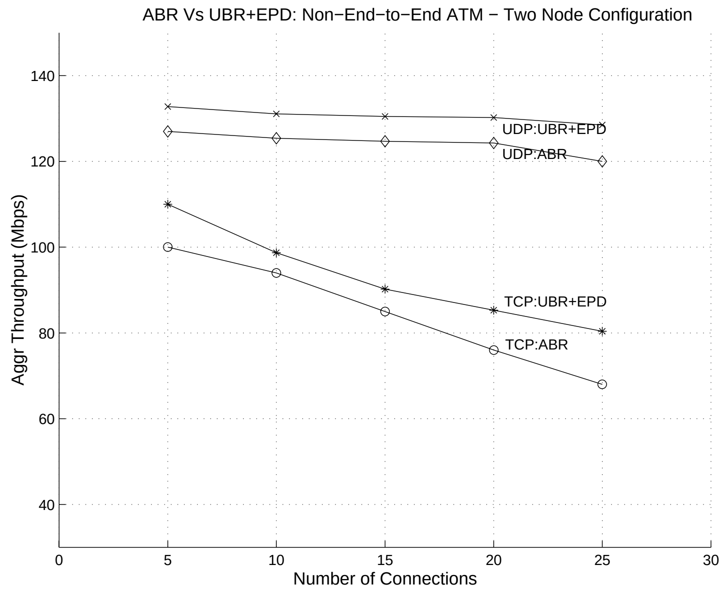
Referring to Figure 5.10(b) it is seen that in the case of of both TCP and UDP the gain is less than 1, but still close to 1, thus implying that in a non-end-to-end scenario especially when there is a single congested node, there is not any gain in using ABR to UBR+EPD except in terms of fairness.

Pacing UDP traffic

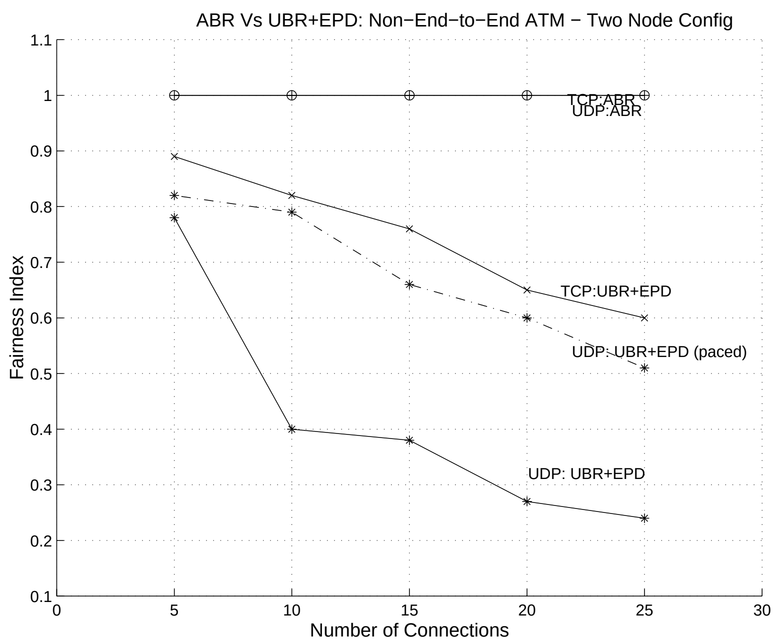
Pacing the UDP traffic to get about the same packet loss ratios as in the case of TCP, only improved the fairness for UDP over UBR+EPD. Hence only the fairness curve is shown for the paced UDP traffic.

5.3.2 Two Node Configuration with Multiple Sources Connected to a Router

In the network scenario considered in the previous section only one TCP source was connected to edge router. In a more realistic network scenario, a router carries multiple TCP flows. This experiment is intended to see this effect. Multiple TCP connections multiplexed to one router, cause heavier congestion at the edge device. Unfairness occurs even when ABR is used if the edge device does not implement sophisticated cell dropping schemes. Also throughput is degraded because fragmented packets get across

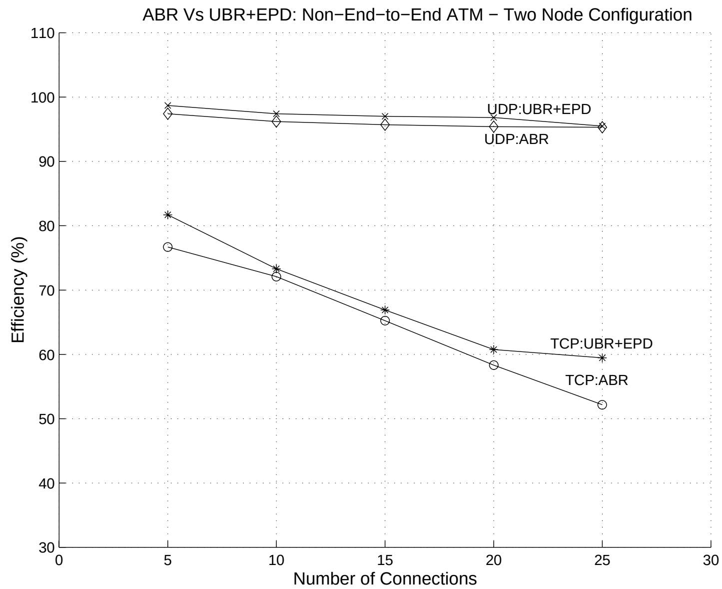


(a)

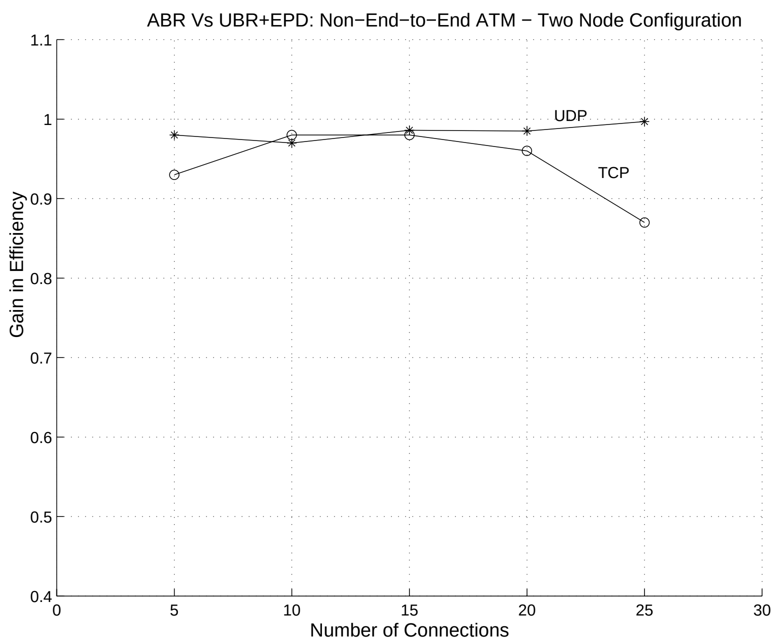


(b)

Figure 5.9: Non-End-to-End Network Scenario: Effect of Number of Connections on Throughput and Fairness Index



(a)



(b)

Figure 5.10: Non-End-to-End Network Scenario: Effect of Number of Connections on Efficiency and Gain

the link and finally get discarded at the destination.

The configuration shown in Figure 4.2 of Section 4.1.1 is used for this experiment. Five TCP flows are multiplexed in router 1 while the other routers carry single TCP flows each. The are five VCs at the ATM level while the total number of TCP/UDP sessions is 9. Thus there is 5:1 loading on the switch and 5:1 loading on router 1.

The results from the simulation on this configuration are presented in Table 5.18.

	Aggr. Goodput 1-5	Aggr Goodput 6-10	Aggregate Goodput (*Mbps*)	Efficiency (%)	FI	PLR	Retrans %
ABR	15.4	88.8	104.2	79.9	0.61	12.3	20.2
UBR+EPD	62.3	57.75	125.6	93.38	0.89	11.9	19.3

Table 5.18: Non-End-to-End ATM Network Scenario - TCP over ABR Vs UBR+EPD: Two Node Configuration with multiple sources through the Router

It is observed that the fairness index is less than 1 even with ABR and is in fact less than the fairness achieved with UBR+EPD. Though ABR attempts to divide the bandwidth equally among all the 5VCs at the ATM level, since the edge device does not have any fair allocation mechanism, the TCP sources 1-to-5 which are multiplexed as one VC get different amounts of throughputs and this contributed to unfairness. Since ABR divides the bandwidth equally, each of the five VCs get a fair-share of 30 Mbps. Hence TCP sources 1-to-5 which are multiplexed into VC1 have only 30 Mbps available. Hence TCP sources 1-to-5 get very low throughput(= 30/5 less the overheads) compared to the other 4 TCP sources *viz.,* sources 6-to-9. On the other hand, with UBR+EPD, all the 10 TCP connections contend for the resources in the switch, each of the connections competes for the OC-3c link bandwidth, hence on average each should get a fair-share of 15 Mbps less the overheads.

So in such a model, UBR+EPD outperforms ABR in terms of achieving better overall efficiency and fairness. Since ABR pushes congestion to the edge device, and since the edge device has no intelligent congestion handling mechanism, in scenarios where router carries multiple flows, ABR provides no benefit relative to UBR+EPD.

5.3.3 Multiple Node Configuration 1

We now evaluate the performance of TCP and UDP over the two ATM services with the Multiple Node Configuration 1 for the ATM subnet in the non-end-to-end ATM network scenario.

The results are summarized in Table 5.19 for TCP and in Table 5.20 for UDP. It is

	Aggregate Goodput ($Mbps$)	Efficiency (%)	FI	CLR(%)	PLR(%)	Retrans. %
ABR	105	80.5	1	1.65	10.3	13.4
UBR+EPD	98.8	73.45	0.89	2.15	12.3	28.75

Table 5.19: Non-End-to-End ATM Network Scenario - TCP over ABR Vs UBR+EPD: Multiple Node Configuration 1

	Aggregate Goodput ($Mbps$)	Efficiency (%)	FI	CLR(%)	PLR(%)
ABR	80.3	61.62	1	55.3	65.31
UBR+EPD	72.6	53.97	0.80	14.4	75.12

Table 5.20: Non-End-to-End ATM Network Scenario - UDP over ABR Vs UBR+EPD: Multiple Node Configuration 1

seen from Table 5.19 that ABR achieves better performance both in terms of efficiency and fairness. ABR tends to outperform UBR+EPD in more complex topologies though the results are fairly similar in simpler topologies.

Effect of Number of Congested Links on the Throughputs of VC1 and VC2

In order to investigate the performance as a function of number of congested links in the ATM networks, for the IP/ATM internetworks, experiments similar to those in section 5.3.3 were repeated. Figures 5.11, 5.12, 5.13, 5.14 plot the throughput, efficiency

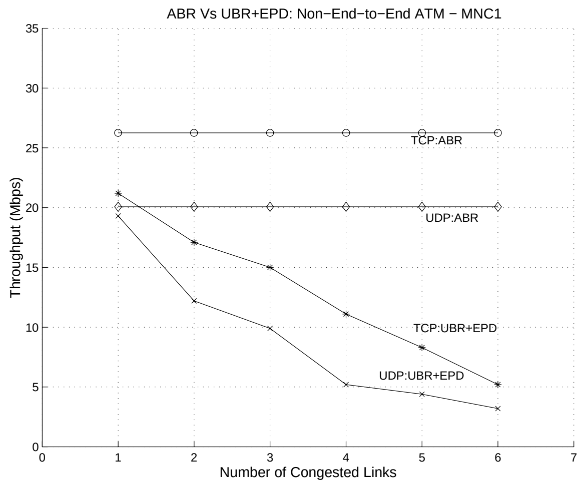


Figure 5.11: Non-End-to-End ATM Scenario - Multiple Node Configuration 1: Throughput Vs Number of Congested Links

for source 1, fairness index and gain of ABR over UBR+EPD as a function of the number of congested links.

It can be seen from these results that ABR clearly outperforms UBR+EPD, and the gain in efficiency with ABR relative to UBR+EPD increases more or less exponentially with the number of congested links in the network. The magnitudes of the gain can be seen to be smaller than in the end-to-end case though, as the efficiency of ABR is degraded by a magnitude of 98.99-80.5 = 18.44 % due to the losses in the edge device.

As the number of congested links increases, UBR+EPD performance degrades for the connections passing though multiple bottlenecks, whereas in the case of ABR, the congestion point is only at the edge device and the ABR mechanism tightly controls the congestion in the switches so the performance is unaffected by the number of hops in the network.

5.3.4 Summary of Non-End-to-End ATM results

- When there is a single congested link, ABR did not provide any performance gain relative to UBR+EPD. Also the performance of ABR degraded as the load on the network increased. Since ABR pushes congestion to the edge device, as the load or

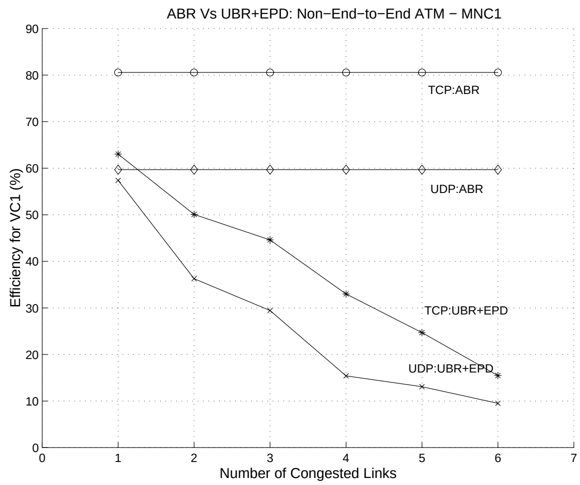


Figure 5.12: Non-End-to-End ATM Scenario - Multiple Node Configuration 1: Efficiency Vs Number of Congested Links

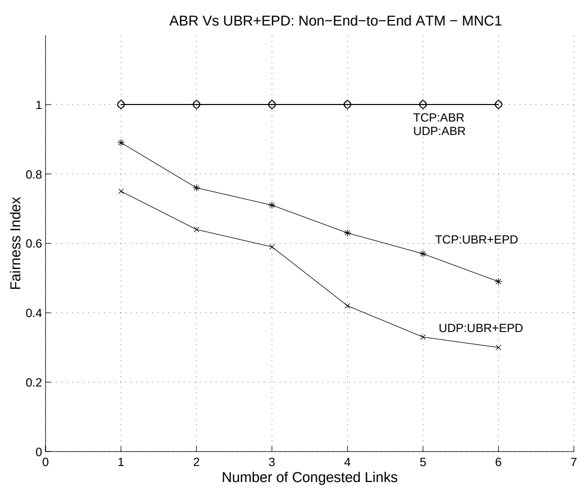


Figure 5.13: Non-End-to-End ATM Scenario - Multiple Node Configuration 1: Fairness Index Vs Number of Congested Links

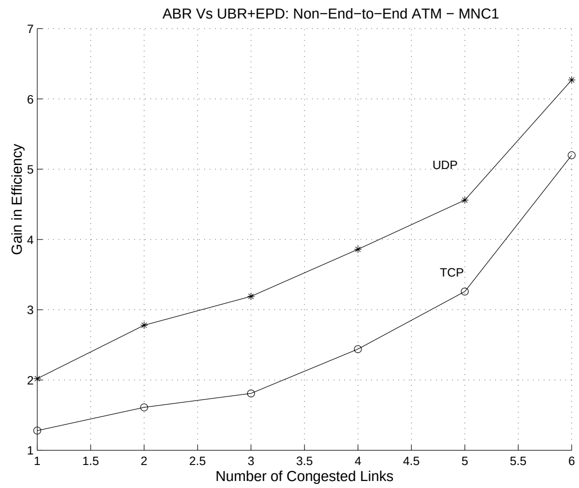


Figure 5.14: Non-End-to-End ATM Scenario - Multiple Node Configuration 1: Gain Vs Number of Congested Links

the number of connections on the ATM network increases and since ABR attempts to divide bandwidth equally among all the connections, the available bandwidth to each of the VCs decreases and hence intensifies the congestion at the edge device. Unless the edge device has sophisticated cell dropping mechanisms, throughput degrades and unfairness occurs.

- In multiple bottleneck scenarios, ABR outperformed UBR+EPD and the performance gain increased as the number of congested links increased. Since with ABR the congestion point is single and is always the edge device, the performance is unaffected by the number of congested links in the network and hence ABR scales well as the size of the network increases. On the other hand, for UBR+EPD, since the congestion point is the switch, as the size of the network increases, the number of congested points increase and the performance of a connection going through the multiple congested switches is degraded severely.

Chapter 6

Conclusions and Future Work

Any study in something as complex as the ABR flow control and it performance assessment in comparison to another service like the UBR is bound to be limited in some respects. In this work, we studied the performance of data traffic sources that use TCP and UDP protocols over ATM networks with ABR service and UBR+EPD service. In particular, a comparison of the ABR and UBR+EPD schemes was made in two scenarios viz., when the network is end-to-end ATM and in an IP/ATM internetworking environment i.e., when the network in non-end-to-end ATM over wide area networks. The key performance metrics were throughput, efficiency, fairness index and gain in efficiency, and the performance assessment was made taking the load, the number of congested links and the location of the congestion point as variable parameters, in both these scenarios.

6.1 Summary of the Results

- In the end-to-end ATM scenarios, ABR always outperformed UBR+EPD both in terms of higher efficiency as well as perfect fairness for TCP. For UDP traffic, however, when the number of congested links was only one, the performance gain with ABR was not that significant relative to UBR+EPD, except in terms of fairness. However, as the network size increased or the number of congested links increased, the performance gain with ABR was conspicuous.

- Further in the end-to-end ATM scenarios, ABR scales well with respect to the load on the network, the number of congested links in the path, and also with respect to the location of the congestion point.

- When there is more than one bottleneck in the network especially when the bottleneck node is downstream of other switches, ABR outperforms UBR, because in UBR+EPD, there is a lot of bandwidth wastage transmitting fragmented packets. Link bandwidth in the upstream nodes is wasted under UBR because cells are admitted into the network, only to be discarded at a bottleneck that may be deep inside the network.

- In non-end-to-end ATM networks, ABR pushes the congestion to the edge device NIC buffers and hence the performance gain decreases compared to the cases where the whole communication path is ATM.

- When number of congested links is only one, there was no performance gain achieved with ABR relative to UBR+EPD. For multiple congested links, ABR outperformed UBR+EPD. As the network size increases the gains achieved with ABR scales up.

- In non-end-to-end ATM networks, the fact that ABR service pushes congestion to the edges of the ATM network while UBR service pushes it inside is an important benefit of ABR for service providers.

- From cost and complexity perspective, the benefits of ABR come at the expense of extra hardware complexity, certain level of overhead into the cell stream due to RM cells, and on-going tuning of a lot of parameters for it to perform well and thus requires additional engineering work.

In general, the choice of either of the service classes, is basically a tradeoff between the costs and the performance that can be achieved. While UBR+EPD is fairly simple, the benefits are limited. The benefits from ABR might be worth the extra cost and complexity.

6.2 Future Work

This work made a comparison of the ABR and UBR+EPD services and the results gave a fairly good insight in to the performance of these two service classes. However, as was mentioned earlier any study is bound to have certain limitations, which we think can be addressed as part of the future work. Some extensions to this work are listed below.

- This work assumed a persistent greedy source for the TCP and UDP traffic, and also a simple ON-OFF source for the background traffic. While this is the worst case for creating congestion and did illustrate the performance implications, real data traffic, such as seen on the Internet is bursty. It could be worth repeating the experiments with such bursty sources and in the presence of a more realistic background traffic, to precisely quantify the gains under more realistic conditions. The relative performance might not change much.

- It was found that in the case of IP/ATM internetworking environments, the relative gain in using ABR over ABR+EPD was less compared to the cases where the TCP sources were directly connected to the ATM network, with the whole communication path being ATM. This was not because of any discrepancies in the ABR mechanism but it was because the routers blocked the ABR feedback from going to the TCP/UDP sources. The benefits of ABR could not be extended to the higher layer protocols, due to lack of signaling of feedback to the real traffic sources. It would be interesting to explore some way of conveying feedback to the TCP sources from the edge routers' ABR NIC or implementing some kind of intelligent cell queueing/dropping such as the RED (Random early detection) in the ABR NIC.

- Further, the results reported for the non-end-to-end ATM case are dependent on the combination of the buffer sizes used at the switch (B_s) and at the edge devices' NIC buffer (B_e). The relative sizing of the buffer pair is a key to the resulting performance. The experiments could be repeated for various B_s and B_e combinations to see how the performance is affected.

- Due to the limitations by the simulator, only small number of TCP connections were considered. Cases with lots of connections need to be studied.

- In this study we have investigated only the UBR+EPD service. It was mentioned in Chapter 2 that UBR service could be enhanced by other dropping policies. We have not investigated the performance of these in this research, as current switch vendors do not yet support these features. Another long term extension is to study the performance improvement that can be obtained by using the enhancements [23] to the UBR service *viz.*, UBR+SPD (Selective Packet Dropping, also called the EPD based on per-VC accounting), UBR + FBA (Fair Buffer Allocation, basically EPD based on per-VC queueing). It could turn out that these enhancements to UBR will result in its outperforming over ABR. This could form an interesting area to explore.

6.3 Contributions

This research work can be used as reference by telecommunication service providers to see the various performance tradeoffs in offering a new service vis_a_vis an already existing service, specifically in making a decision whether to choose ABR or UBR+EPD as the choice of ATM service category for data services.

Bibliography

[1] Douglas Comer, "Internetworking with TCP/IP - Principles, Protocols, and Architecture," Volume 1. Prentice Hall, Englewood Cliffs, New Jersey, 1995.

[2] W. Richard Stevens, "TCP/IP Illustarted, Volume 1: The Protocols," Addison-Wesley Publishing Company, Inc., 1994

[3] Martin De Prycker, "Asynchronous Transfer Mode: Solution for Broadband ISDN" 3rd ed. Prentice Hall, 1995.

[4] William Stallings, "High-Speed Networks: TCP/IP and ATM Design Principles", Prentice Hall, 1998.

[5] F. Ansari, "Adapting TCP/IP to ATM", M.S Thesis, University of Kansas, September 1996.

[6] A. Romanow and S. Floyd, "Dynamics of TCP traffic over ATM Netwroks ", IEEE Journal on Selected Areas in Communications, Volume 13, May, 1995.

[7] Stephen Keung, "Degradation in TCP performance under cell loss", ATM_FORUM/94-0490, 1994.

[8] Kai Yeung Siu et al, "TCP performance over ABR and UBR services in ATM Networks", International Phoenix Conference on Computers and Communicationsi, IPCCC'96, March 1996

[9] Helen Chen, Chien Fang and Jim Hutchins. "A simulation study of TCP performance in ATM networksi", Proceedings of Globecom'94, pages 1217-1223, November 1994.

[10] The ATM Forum Technical Committee, "Traffic Management Specification Version 4.0.," af-tm-0056.000, April 1996.

[11] J. Postel, "Transmission control protocol. Request for Comments (Standard) RFC 793", Internet Engineering Task Force, September 1981

[12] R. Braden, D. Borman and V. Jacobson, "TCP extensions for high performance," RFC 1323, Internet Engineering Task Force, May 1992.

[13] M. Mathis, J. Madhavi, S. Floyd, A. Romanow,"TCP Selective Acknowledgement Options," Internet RFC 2018, October 1996.

[14] Van Jacobson, "Congestion Avoidance and Control", ACM Computer Communication Review, 18(4):314-329, August 1988. Proceedings of the Sigcomm '88 Symposium in Stanford, CA, August, 1988

[15] B. Ewy, J. B. Evans, V. Frost, G. J. Minden, "TCP and ATM in Wide Area Networks ", Gigabit Network Workshop 1994.

[16] E. Klovning, K. Moldeklev and O. Kure. "TCP/IP behavior in a high-speed local ATM network environmenti", Proceedings 19th Conference on Local Computer Networks, pages 176-185, October 1994.

[17] R. Morris and H. T. Kung, "Impact of ATM Switching and Flow Control on TCP performance: Measurements on An Experimental Switch", GLOBECOM '95 Number 2, pp. 888-892

[18] Raj Jain, Rohit Goyal, Shiv Kalyanaraman and Sonia Fahmy,"Performance of TCP over UBR and buffer requirements," ATM FORUM/96-0518, 1996

[19] Shiv Kalyanaraman, Raj Jain, Sonia Fahmy, and Rohit Goyal "Performance and Buffer Requirements of Internet Protocols over ATM ABR and UBR services", IEEE Communications Magazine, January 1997 .

[20] S. Jagannath, N. Yin " End-to-End Traffic Management in IP/ATM Internetworks", ATM Forum Contribution 96-1711, December 1996.

[21] Sam Manthorpe and J. Le Boudec, " A comparison of ABR and UBR to support TCP traffic" DI Technical Report 97/224, April 1997.

[22] Partho Prathim Mishra, "Throughput degradation for TCP over ATM in the presence of Traffic Policing," TCGN Gigabit Networking Workshop 1995.

[23] Roch Guerin, and Juha Heinanen, "UBR+ Enhancements", ATM FORUM 97-0015, February 1997.

[24] MIL3, Inc., OPNET Modeling Manual Volume 1, 1996.