PERFORMANCE EVALUATION OF CONGESTION CONTROL MECHANISMS IN ATM NETWORKS

Paper Number 111

Abstract

Future high-speed computer communication networks are expected to utilize Asynchronous Transfer Mode (ATM) technology. In order to broadly deploy ATM-based networks, numerous congestion control mechanisms are presently being developed by the industry and research community alike. In this paper, we propose a general method to evaluate the performance of a given congestion control scheme by observing the throughput of a connection. This method was used and shown to be effective in testing the performance of credit-based flow control implemented in a high speed network testbed. We present the results from experiments conducted in a wide area communication environment.

1 Introduction

Communication networks for the mid-1990s and beyond are expected to support a wide variety of services, from the transmission of voice, video and data to new, commercially attractive services such as high-definition TV, high quality videophony, high speed data transfer, video on demand, etc. [1] [2]. This concept is often referred to as Broadband Integrated Services Network, or BISDN.

BISDN is expected to utilize Asynchronous Transfer Mode (ATM) technology. ATM technology has been chosen by numerous standards committees on the basis of its ability to integrate user traffic from existing applications onto a single user-network interface and to facilitate multi-media networking between high-speed devices at multi-gigabit speeds [3].

As in any packet-switched communication network, the future ATM-based BISDN should deploy certain mechanisms in order to avoid network congestion. In general, the need for congestion control arises whenever the load offered to a network connection is larger than the available bandwidth. In order to understand the congestion control problem in ATM networks, we introduce the ATM layer service classes defined by the ATM Forum [4]:

1. **Constant Bit Rate (CBR) :** This class is used for emulating circuit switching. The cell rate is constant and a Cell Loss Ratio (CLR) requirement is specified. Examples of applications that use CBR are telephone, video conferencing, and television.

2. **Variable Bit Rate (VBR) :** This class allows users to send at a variable rate (bursty traffic). The average cell rate, CLR, and cell transfer delay (CTD) are specified. An example of real-time VBR is interactive compressed video, while that of nonreal-time VBR is multimedia e-mail.

3. **Available Bit Rate (ABR):** This class is designed for traditional data traffic such as file transfer and e-mail. The standard does not require the CTD and CLR to be guaranteed by the network. The Minimum Cell Rate (MCR) is specified for ABR. However, most ABR Virtual Circuits (VCs) will request an MCR of zero. Those with higher MCR may be denied connection if sufficient bandwidth is not available at the call setup time.

4. **Unspecified Bit Rate (UBR):** This class is designed for 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 (no connection admission control) and not policed for their usage behavior. During congestion, cells are lost but sources are not expected to reduce their transmission rate.

Of all the service classes described above, only ABR traffic responds to congestion information fedback from the network. Therefore, congestion control mechanisms are only focused on the ABR service class. For CBR and VBR traffic, shaping and policing mechanisms can be used. Also, since the UBR VCs do not slow down during congestion, it is expected that the network should provide a certain degree of isolation in order to protect the quality of other service classes.

The rest of this paper is organized as follows: Section 2 introduces a methodology for testing the performance of different congestion control schemes; Section 3 presents a case study using this methodology; and Section 4 draws a conclusion for the paper.

2 Performance Testing Methodology

2.1 Testing Model

Figure 1 shows the network model we propose to use for testing the congestion control mechanisms.

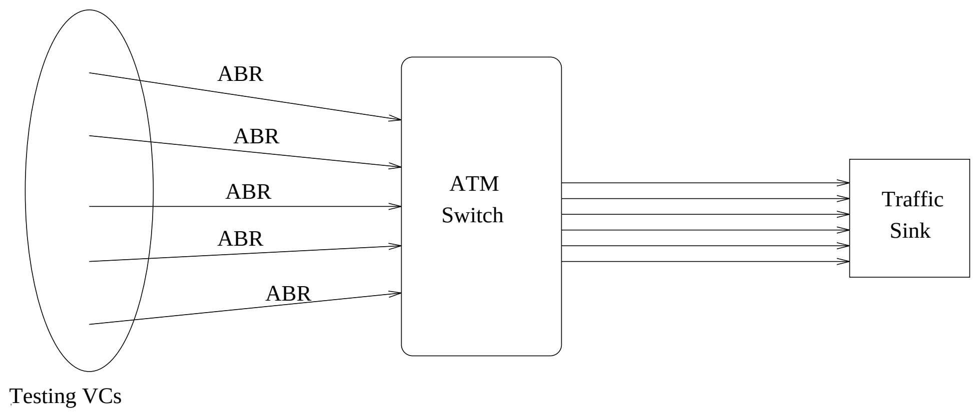

Figure 1: Testing Model

In this model, a group of ABR traffic streams are used as testing VCs. The CBR and VBR service classes are not included in the testing VCs because they all require certain degree of average throughput and do not respond to network congestion information. We also assume all ABR VCs always attempt to transmit as fast as possible (blasting traffic mode). Therefore, the link is always busy, i.e., there is always a cell to send in each ATM cell slot. The blasting traffic mode allows us to measure the maximum aggregate network throughput under different congestion control mechanisms.

2.2 Testing Metrics and Performance Criteria

When VCs are contending for bandwidth and the aggregate cell arrival rate to a node exceeds the link capacity, cells can be lost due to buffer overflow. Cell losses may cause the retransmission of upper layer packets, which imposes a higher load on the network and further deteriorates throughput. In our methodology, we use throughput as the main performance metric. Moreover, for any effective congestion control scheme, the following criteria should be met:

Fairness: The bandwidth left over by CBR and VBR VCs should be equally shared by those ABR VCs with same MCRs, i.e.,

$$B_i = MCR_i + \frac{A - \sum_{j=1}^{N} MCR_j}{N}, i = 1, 2, 3......N, \tag{1}$$

where B_i is the fair share of bandwidth for ith ABR VC, A is the total available bandwidth left over by CBR and VBR traffic, and N is the total number of active ABR VCs.

Throughput Degradation: When congestion occurs, the network should respond fast enough so that the aggregate throughput degradation be kept to a minimum. This means the aggregate throughput should approach the link capacity, i.e.,

$$\sum_{i=1}^{N} B_i \rightarrow C, \tag{2}$$

where N is the total number of active VCs, B_i is the throughput for the ith VC, and C is the link capacity.

2.3 The Role of TCP and UDP

Most of the existing ATM networks utilize a TCP/IP over ATM architecture. It is chosen mainly due to the infancy of ATM technology as well as the popularity of TCP/IP protocols. This architecture offers a way to emulate courteous and non-courteous traffic streams. Specifically, TCP is viewed as a courteous protocol because its window mechanism and slow start algorithm make it back off when network congestion is high. UDP, on the other hand, is viewed as non-courteous because it has no back-off operation and continues to transmit even in high-traffic conditions [6].

These characteristics of TCP and UDP are very useful in testing the effectiveness of congestion schemes. As an illustration, imagine an ATM network without a congestion control mechanism; also assume two testing VCs, both configured as zero MCR ABR VCs, one used for transmitting UDP traffic and the other for transmitting TCP traffic. This traffic scenario emulates an unfair situation. Under blasting mode, UDP keeps sending traffic as fast as possible, causing congestion; while TCP detects the congestion and reduces its traffic rate. An effective congestion control scheme should be able to prevent this and fairly allocate the bandwidth between the two traffic streams.

2.4 Testing Procedure

We summarize the testing procedure as follows:

1. Configure the ATM network into various scenarios according to the testing model.

2. Configure TCP and/or UDP traffic streams as ABR testing VCs in order to emulate courteous and non-courteous traffic.

3. Let these ABR VCs send traffic through the network in blasting mode and measure the throughput for each VC. Throughput should be measured at both the transmitting and receiving ends in order to locate bottlenecks, if any.

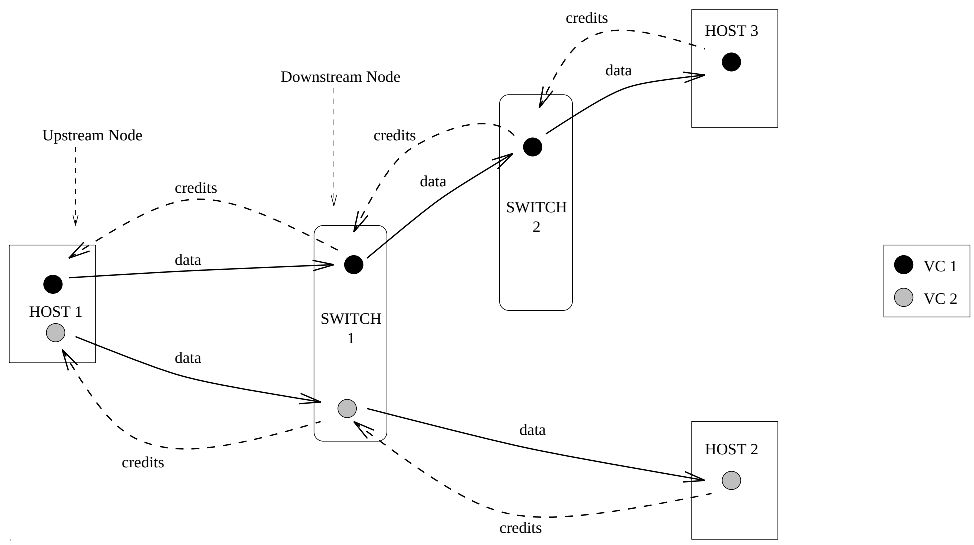

Figure 2: Link-by-link credit-based congestion control example

4. Analyze the throughput results based on the testing criteria. The level to which these criteria are met indicates the effectiveness of the congestion control scheme.

3 Case Study: Application to Credit-Based Congestion Control

In this section, we will apply the previously described methodology to evaluate the performance of credit-based congestion control in a high-speed ATM testbed.

Credit-based congestion control consists of a sliding window algorithm, illustrated in Figure 2. In order to transmit a cell on a given VC, an upstream node must have credits available. Periodically, a downstream node sends credit updates to the upstream node, indicating the availability of buffer space for receiving data on a VC [5]. The credit count is decremented for each transmitted cell. This procedure is followed independently for each link, and hence we sometimes refer to this algorithm as link-by-link congestion control. Once a node runs out of credits, it is no longer allowed to transmit any cells. In this fashion, when congestion occurs, cells will not be acknowledged and upstream nodes will eventually run out of credits.

3.1 Experiment Set-Up

In testing the performance of credit-based congestion control, we used the set-up shown in Figure 3. Hosts A and B are two DEC 3000 workstations located

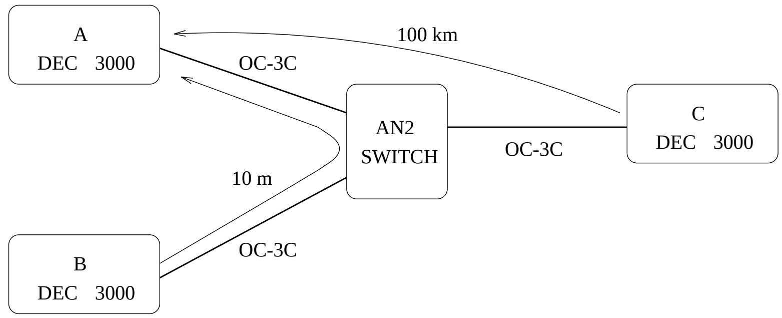

Figure 3: Experiment Set-up

in the Telecommunications and Information Sciences Laboratory (TISL), at the University of Kansas. They both access a DEC AN2 local area ATM switch at OC-3 rates (155.52 Mbps). Credit-based congestion control is an important feature of the AN2 switch. Host C is also a DEC 3000 workstation, located in the Sprint Technology Integration and Operation Center (TIOC), Kansas City, at a distance of approximately 100 km from the switch, and connected to it at OC-3 rates.

Note that, although the nominal OC-3 rate is 155.52 Mbps, the theoretical maximum achievable bandwidth is 135.102 Mbps once we exclude overhead. Experimental results for throughput may be even lower due to processes running on the hosts (e.g., daemons).

In all of our experiments, host A is the receiving host, and B and C are transmitting hosts. Testing traffic streams run through permanent virtual circuits (PVCs) which have been set up for these experiments.

3.2 Credit Allocation

The total number of credits for a given VC determines the amount of cell buffer space that will be allocated for that VC on each node. The number of credits should be large enough not to impact throughput under non-congested states.

Since all links operate at 155.52 Mbps and ATM cells have a fixed size of 53 octets, the number of credits to be allocated can be calculated as:

$$Number\ of\ Credits = \frac{Round\ Trip\ Delay \times\ 155.52 Mbit/s}{53 \times\ 8 bits} \tag{3}$$

For the long-distance connection (C to A), propagation delay dominates the round trip delay. For the local-area connection (B to A), the distance between the two hosts is approximately 10 m. Therefore, in this case the propagation delay is negligible and credit processing delay at the hosts and switch is the predominant component of round trip delay.

A simple experiment can be run in order to estimate the credit processing delay.

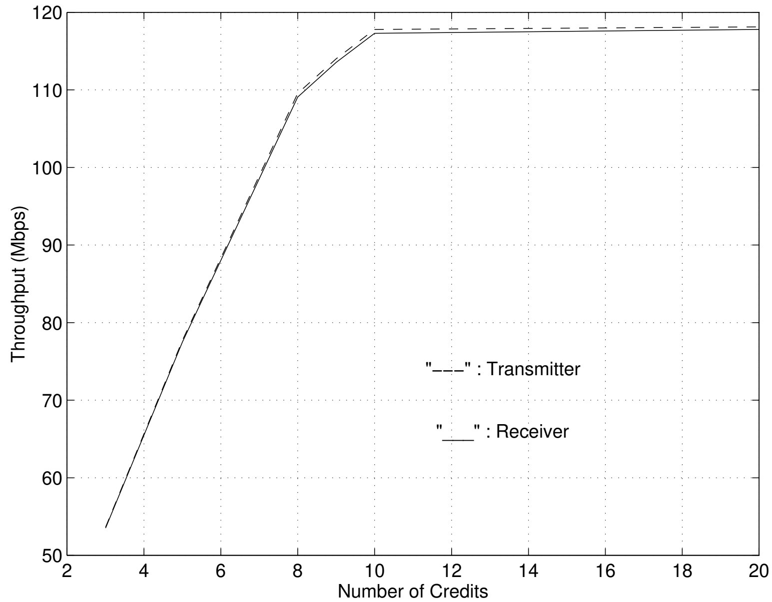

Figure 4: Throughput on B to A connection under Flow Control, with TCP traffic. Data points taken over 10 repetitions of the experiment.

A PVC is set up between hosts B and A, without any cross traffic and making sure the network has no load. TCP traffic is generated and sent in blasting mode from host B to host A, and throughput is measured with varying number of credits. The results are shown in Figure 4.

The results indicate that the maximum throughput is achieved at about 10 credits. Notice that allocating more credits than required will not impact throughput, as can be seen in Figure 4. However, as we increase the number of credits, we allow for higher queueing delay; we also incur the cost of additional memory. Assuming a negligible propagation delay, we can therefore estimate that 10 credits are needed to accomodate the credit processing delay.

For the long-distance connection, we know the distance between hosts A and C is 100.20 km and the speed of light in fiber is about $0.69 \times 3.0 \times 10^8$ meters/s. Using Equation 3 , we calculate the number of credits needed to account for propagation delay between C and A as 355. To these, we must add the 10 credits needed to accomodate the processing delay. Therefore, the necessary number of credits for the long distance connection is approximately 365.

3.3 Throughput Measurement Tool

As discussed in Section 2, we need to generate TCP and UDP traffic in a blasting mode for the Testing and Contending VCs. In order to generate these and measure the throughput on each PVC, the TTCP program was used. TTCP is

	Congestion Control		No Congestion Control	
	Transmitter	Receiver	Transmitter	Receiver
C to A (TCP)	58.5 Mbps	58.5 Mbps	57.1 Mbps	57.0 Mbps
B to A (TCP)	56.2 Mbps	56.3 Mbps	40.4 Mbps	40.5 Mbps
Aggregate Throughput		114.8 Mbps		97.5 Mbps

Table 1: Individual and aggregate throughput for 2 contending TCP traffic streams.

	Congestion Control		No Congestion Control	
	Transmitter	Receiver	Transmitter	Receiver
C to A (UDP)	68.6 Mbps	44.5 Mbps	129.5 Mbps	4.1 Mbps
B to A (UDP)	67.4 Mbps	44.2 Mbps	127.6 Mbps	5.7 Mbps
Aggregate Throughput		88.7 Mbps		9.8 Mbps

Table 2: Individual and aggregate throughput for 2 contending UDP traffic streams.

a benchmarking tool for measuring the throughput between two systems using the UDP or TCP protocols. The program was created at the US Army Ballistics Research Lab (BRL) and is in the public domain.

3.4 Numerical Results

Table 1 through Table 3 summarize the results of a set of experiments testing the performance of Credit-Based Congestion Control. All individual throughput values presented in the tables are averages taken over a repetition of ten experiments.

From Table 1, we can observe that when two courteous traffic streams compete for the same link, the aggregate throughput without congestion control is almost 15% lower than that with congestion control. The reason for this is that while TCP's backoff mechanism prevents severe throughput degradation, it cannot prevent cell losses from happening at the switch.

From Table 2, it's found that without congestion control, none of the two UDP traffic streams can get good throughput, and the aggregate throughput is only 9.8Mbps; with congestion control, although there is still some cell loss in the transmitting hosts due to the non-courteous nature of UDP, the aggregate throughput is much better.

From Table 3, it's found that without congestion control, UDP takes hold of most of the bandwidth, causing an unfair situation; with congestion control, fairness is achieved and aggregate throughput is also improved.

	Congestion Control		No Congestion Control	
	Transmitter	Receiver	Transmitter	Receiver
C to A (TCP)		55.7 Mbps		9.5 Mbps
B to A (UDP)	69.7 Mbps	41.2 Mbps	120.4	78.5 Mbps
Aggregate Throughput		96.9 Mbps		88.0 Mbps

Table 3: Individual and aggregate throughput for a TCP stream flowing from C to A and an UDP stream from B to A.

4 Conclusion

In this paper, a method for testing the performance of congestion control schemes for ABR traffic in ATM networks is presented. A set of experiments was conducted on a high-speed ATM testbed and used to evaluate the effectivenss of ATM level Credit-Based congestion control. Experimental results indicate that a Credit-Based congestion control scheme works effectively in minimizing aggregate throughput degradation during congestion and ensuring a fair allocation of the available bandwidth in both wide-area and local area environments.

References

[1] Martin de Prycker, *Asynchronous Transfer Mode - Solution for Broadband ISDN*, 2nd edition, Ellis Horwood, 1993.

[2] D. V. Batorsky, D. R. Spears and A. R. Tedesco, "The Evolution of Broadband Network Architectures," *Proceedings of Globecom,* 1988, pages 367-373.

[3] The ATM Forum Technical Committee, *User-Network Interface (UNI) Specification Version 3.1*, 1994.

[4] ATM Forum Traffic Management Group, *Draft Version 2.0 of ATM Forum Traffic Management Specification Version 4.0*, ATMF 95-0013R1, 1995.

[5] The ATM Forum Technical Committee, *Flow Controlled Virtual Connections - Proposal for ATM Traffic Management*, September 1994.

[6] Uyless Black, *TCP/IP and Related Protocols*, 2nd edition, McGraw-Hill, 1995.