AIIM

Association for Information and Image Management

1100 Wayne Avenue, Suite 1100
Silver Spring, Maryland 20910

301/587-8202

Centimeter

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 mm

1 2 3 4 5

Inches

1.0 1.1 1.25 1.4 1.6 1.8 2.0 2.2 2.5 2.8 3.2 3.6 4.0 4.5 5.0 5.6 6.3

1 of 1

2.

SAND--94-8728C
Conf-9411102--1

A SIMULATION STUDY OF TCP PERFORMANCE IN ATM NETWORKS

Chien Fang, Helen Chen, Jim Hutchins

Infrastructure and Networking Research Department
Sandia National Laboratories
Livermore, California

ABSTRACT This paper presents a simulation study of TCP performance over congested ATM local area networks. We simulated a variety of schemes for congestion control for ATM LANs, including a simple cell-drop, a credit-based flow control scheme that back-pressures individual VC's, and two selective cell-drop schemes. Our simulation results for congested ATM LANs show the following: (1) TCP performance is poor under simple cell-drop, (2) the selective cell-drop schemes increase effective link utilization and result in higher TCP throughputs than the simple cell-drop scheme, and (3) the credit-based flow control scheme eliminates cell loss and achieves maximum performance and effective link utilization.

Introduction

Asynchronous Transfer Mode (ATM) [16] has emerged as a standard which offers integration of different classes of service as well as scalability in terms of network size and speed. While the early focus on ATM was mainly on wide-area networks, the driving force behind this new technology is increasingly coming from local-area applications. The need for an ATM local area network (LAN) to support existing protocols (e.g., TCP/IP [6]) transparently is key to its adoption in the local-area environment.

In traditional shared medium LANs, congestion control for the LAN is handled by the medium access control (MAC) protocol. Once a network device gains access to the shared medium, it has the full bandwidth to itself. Other contending network devices must wait until the shared medium becomes free. In ATM LANs, the MAC protocol is removed, and as a result the congestion control it provides is also removed. Each network device can transmit at the full link bandwidth, and can thus potentially cause congestion at some output ports of an ATM switch.

For constant-bit-rate (CBR) and variable-bit-rate (VBR) [8] services, an ATM network provides guarantee to the quality of service (QoS) in terms of a set of parameters specifying bandwidth, delay, and jitter bounds. For these services, congestion control is administered through admission control and bandwidth allocation. If an ATM network cannot deliver the resources requested by a connection request, the connection request will be rejected at call setup time. For "Best-Effort", or available-bit-rate (ABR) service, the resource requirements are not specified. Therefore, admission control and bandwidth allocation cannot be used to provide congestion control for ABR services.

The problem of cell loss as a result of congestion in an ATM network is well-recognized [2, 9, 7]. A number of congestion control schemes have been proposed for ATM networks. The Forward Explicit Congestion Notification (FECN) scheme [20] and the Backward Explicit Congestion Notification (BECN) scheme [15] are two examples of reactive rate-based schemes. In the FECN scheme, a cell is marked at the point of congestion and then forwarded to its destination as usual. After receiving the marked cell, the destination node send an explicit congestion notification to the source to throttle back its transmission rate. The BECN scheme differs from the FECN scheme in that congestion notifications are sent directly back to the source by the switch experiencing congestion. Both the BECN and FECN schemes can achieve a low cell-loss rate by provisioning sufficient buffers. There are also two proposed credit-based link-by-link flow control schemes, the AN2 flow control [18] and the Flow-Controlled Virtual Channels (FCVC) [12] scheme. These two schemes are based on virtual-circuits (VC) and require per-VC buffering. The basic idea in the two schemes is that a sender will not transmit a cell unless it knows that a buffer is available at the receiver to hold the cell. Credit-based per VC flow control schemes provide fast feedback and are especially effective in handling transient congestion in high-speed data networks. They also guarantees zero cell loss in the absence of hardware errors.

Simulation studies on the above congestion control schemes have been presented [11, 13, 15, 20]. These studies all provide performance measures at the ATM cell level (e.g., cell throughput and cell loss rate). However, the effect of cell loss on the performance of higher-level protocols (e.g. TCP, IPX, DECnet etc.) is still not well-understood. Recently, Romanow et al. [17] have presented a simulation study of TCP traffic over ATM networks. Their simulation model is simplistic in a number of ways. First, parameters such as TCP packet processing time, cell segmentation and reassembly time, and switch delays are ignored. By ignoring these parameters, their simulation results are likely to be over-optimistic. Second, their model only reports TCP throughput values. Our simulation model is much more realistic and models all of the above parameters as well as reports throughput, round-trip delay, cell loss rate, TCP retransmission rate, and effective link utilization.

This paper will present a simulation study on TCP performance in a congested ATM LAN. A number of congestion control schemes are simulated: (1) simple cell-drop scheme, where cells arriving at a full queue are simply dropped, (2) AN2 flow control algorithm [18], a per-VC per-hop credit-based scheme, (3) selective "drop-tail" scheme, where all the cells which belong to the same TCP packet and which arrive after the first discarded cell are also discarded, and (4) selective "drop-whole" scheme, where all cells belonging to a "corrupted" TCP packet are dropped.

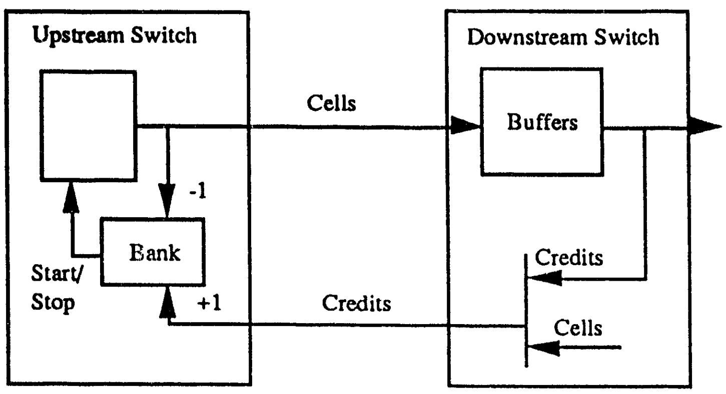
The remainder of this paper is organized as follows. Section II provides some background information and related work. Section III presents the simulation study. Section IV gives our conclusion and describes future work plans.

Background

AN2's Flow Control Algorithm

The AN2 flow control algorithm proposed by DEC [18] is a per-hop per-VC, credit-based scheme to be used by any switch which has per VC buffer management. As depicted in Figure 1, the sender maintains a credit balance for each VC initialized to the buffer size at the receiver. For each cell forwarded, the sender decrements the credit balance for the VC by one. No cell will be sent when the credit balance reaches zero.

For Each VC:



Need one link round trip of buffering per VC

Figure 1. DEC's AN2 Credit-based Flow Control Algorithm

When the receiver has successfully forwarded a cell, a credit is returned to the sender and its credit balance is incremented. Credits are piggybacked onto either a data cell or a null cell traveling in the opposite direction. The AN2 scheme uses the VPI field and the high-order 4 bits of the VCI field of the ATM cell header to carry the credit information. By carrying credits in the cell header, AN2's flow control scheme does not incur bandwidth overhead. The buffer requirement for this scheme is equal to the product of the targeted VC bandwidth and the round trip propogation delay of the link between the sender and the receiver. The targeted VC bandwidth can be chosen to be less than or equal to the link bandwidth.

Simulation Tool

Our simulation tool is based on the MIT Network Simulator (NetSim) [14]. NetSim is an event-driven simulator composed of various component modules that send messages to one another. We have built upon the NetSim package by adding various ATM related components. These components include an ATM switch component, a SONET OC-3c link component, and an ATM host component. The ATM host component performs ATM adaptation (AAL5) [5] and segmentation and reassembly (SAR) of TCP/IP packets. The ATM switch module used for this paper models DEC's AN2 switch. It is a 16x16 input buffered crossbar switch based on ATM. Each input port has 16 queues, one per output port. Each queue contains a list of VC queues destined for the output port. The output arbitration mechanism uses parallel iterative matching [1], which together with per-VC buffer management can prevent head-of-line blocking.

To create a congested ATM LAN, we use a network configuration as shown in Figure 2.

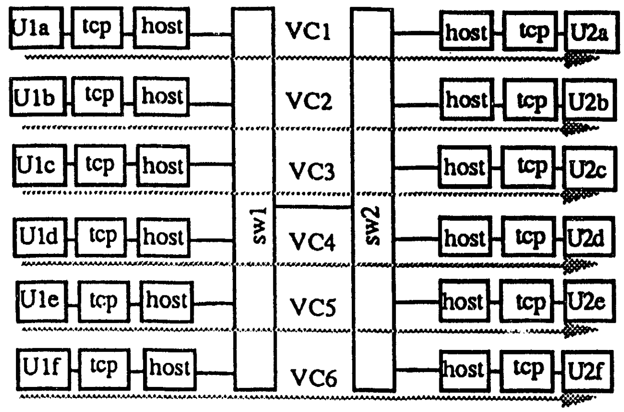


Figure 2. Simulated ATM network configuration of six concurrent TCP sessions

On the sending side, the user modules (U1's) generate data for TCP modules (tcp), which performs data segmentation and forms TCP packets, which in turn are passed onto host adapters for IP routing, AAL5 and SAR processing. The two ATM switches performs cell switching between their input and output ports. On the receiving side, cells are reassembled and passed on to the TCP and user modules (U2's). By running 6 concurrent TCP sessions, we created a congested link between the two ATM switches.

The traffic model for TCP provided by NetSim is a greedy traffic source; each TCP connection tries to send as much data as fast as it can. The TCP module in NetSim is based on Berkeley Standard Distribution (BSD) version 4.3 which uses the slow-start congestion control and is adaptive to network conditions [10]. However, instead of employing the BSD slow-timer mechanism for TCP retransmission, NetSim adopted a fast TCP retransmission algorithm. This algorithm retransmits TCP packets immediately after they are timed out; if adopted by workstations, the fast TCP retransmission will incur extra processing overhead to service timer interrupts. Historically, workstations are slower; the slow-timer mechanism is implemented in order to avoid extra processing overhead. The slow-timer expires every 500 ms; at which time, TCP will service all TCP packets pending for retransmission. Today's workstations are much faster; they can handle the processing overhead of the fast TCP retransmissions. However, the need for the fast TCP retransmission algorithm has not been recognized. Similar to advances in workstation, today's networks have also become faster and more reliable. Under reliable network, TCP retransmissions are rare exceptions. Therefore, the slow-timer survived and are still used by most BSD-based workstations. We believe that the TCP performance obtained by workstations with the slow-timer mechanism would be much worse than our simulation results.

Simulation Paramenets

Simulation Parameter	Value
TCP Window Size	64 KByte
TCP Maximum Segmentation Size (mss)	8 KByte
TCP Processing and OS Overhead Delay	300 µS
Host Segmentation and Reassembly Delay	200 nS
Switching Delay	4 µS
Switch Per VC Queue Size (All Other Simulations)	256 cells
Switch Per VC Credit Grants (Flow Control)	5 Cells
Link Serialization Delay at OC-3c Speed	2.75µS
Link propagation Delay (100m)	500 nS

Table 1. Run-time Simulation Parameters

We selected our simulation parameters based on experimental measurements obtained by DEC and Sandia [1, 4] and they are listed in Table 1.

Note that because of the finite resolution of the simulator clock, the simulated OC-3c link speed is actually 53x8/2.75mS = 154.2 Mbps. Taking into consideration the ATM overhead, the maximum achievable throughput is 154.2x(48/53) = 139.5 Mbps. Also note that the buffer requirement for the AN2 flow control is 5 cells per-hop per VC and it is the amount of memory required to accommodate the round-trip propagation delay as well as the end processing delay of a hop. In the absence of flow control, the per VC buffer requirement is much larger; it must be large enough to hold cells of a complete TCP packet in order for a TCP session to proceed successfully. At 8 Kbyte per packet, the per VC queue size must be equal to or greater than 171 cells. We have chosen a queue size of 256 cells because it is the minimum size that is a power of 2 and allows reasonable TCP performance in our simulation. We have also used 256 cells as the queue size for the slelctive cell drop simulations. Therefore, in comparing their simulation results, we must keep in mind that the per VC buffer requirement of the credit flow control scheme is 51times less than required by other schemes.

Simulation Results

Baseline configuration

To determine TCP performance in the absence of congestion, we simulated the baseline configuration. The baseline configuration consists of only one active TCP session in Figure 2 (e.g., VC1). The throughput values for the simple cell-drop and the AN2 flow control scheme are plotted against the simulation time and are presented in Figure 3. Note that the AN2 congestion control algorithm does not effect TCP throughput performance in the absence of congestion.

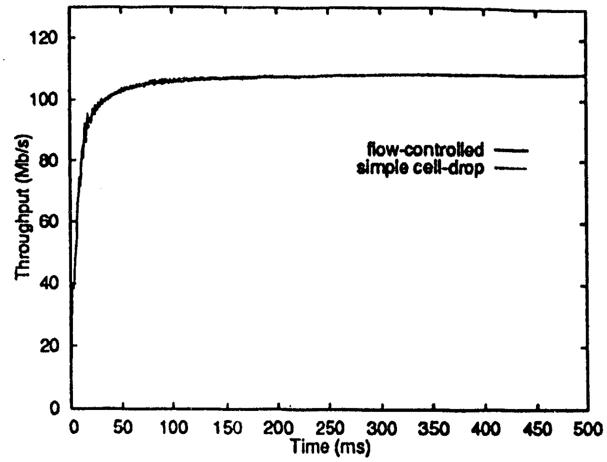


Figure 3. Baseline Throughput Performance

Simple Cell-Drop and The AN2 Flow Control

Figure 4 plots the throughputs for the simple cell-drop and the AN2 flow control schemes as a function of simulation time. The upper group of curves represent the throughputs of the six TCP connections with AN2 flow control in place. As shown, AN2 flow control scheme allows the bandwidth to be shared evenly among the six TCP sessions; each session has roughly one-sixth of the OC-3c bandwidth, 23.3 Mbps. Since each session is able to used one sixth of the link bandwidth, the AN2 flow control scheme demonstrated full link utilization.

The lower group of curves in Figure 4 represent the throughputs for the six TCPsessions for the simple cell-drop scheme. It can be seen that there is a 36% TCP throughput degradation relative to the flow-controlled throughputs. In addition, note that the per VC buffer requirement of the non-flow-controlled simulation is 256 cells and it is much larger than the buffer requirement of 5 cells for the AN2 flow control scheme. Moreover, the variance of the throughputs of the six TCP sessions for the simple cell-drop scheme is higher, showing a lesser degree of fairness in bandwidth sharing than the flow-controlled case.

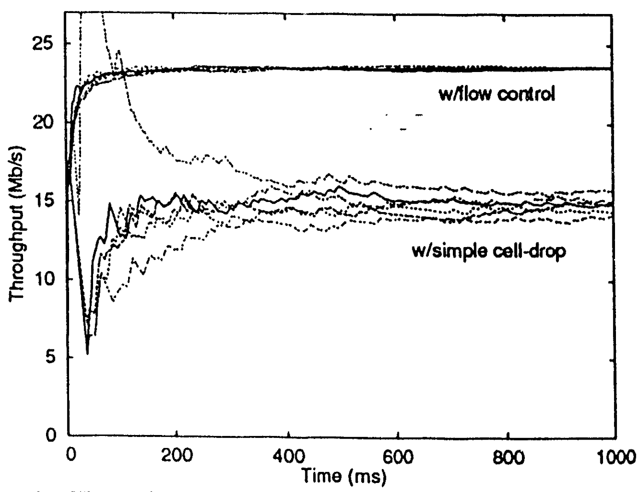


Figure 4. Throughputs for simple cell-drop and flow control schemes

The fairness in the credit-based flow control scheme is further demonstrated by a simulation run in which the TCP packet size (or maximum segment size, mss) for one of the six sessions is reduced from 8 Kbytes to 64bytes. By using a really small TCP mss, the protocol-processing overhead at the two end-stations becomes the dominant factor in determining throughput, resulting in a severely reduced data rate. As shown by Table 2, the maximum throughput achievable by the 64 byte session in the absence of congestion is 0.758 Mbps. Because a throughput of 0.758 Mbps is much less than one sixth of the OC-3c bandwidth, the credit-based flow control mechanism allows the session to retain much of its bandwidth and throttled the other five sessions to share the remaining bandwidth. This is an example where the per VC buffer management can eliminate undesireable interactions between different VCs.

	TCP Segment Size					
	64 B	8KB	8KB	8KB	8KB	8KB
Thruput (Mbps)	0.754*	27.95	28.19	27.79	28.15	27.88

* The standalone throughput for this session is 0.758 Mbps

Table 2. Simulation throughput of six TCP sessions with 8 Kbyte and 64 byte packet sizes.

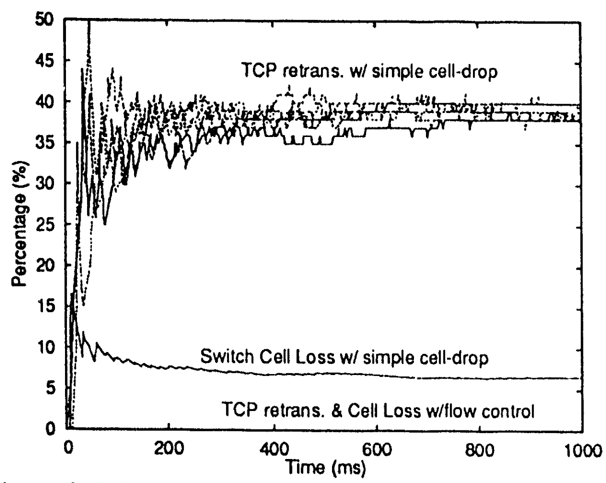


Figure 5 Cell loss and TCP Retransmission percentage

Figure 5 plots the cell-loss rate and the TCP retransmission rate as a function of simulation time. With the AN2 flow control, the simulation results show zero cell-loss and zero TCP retransmission. With the simple cell-drop scheme, the TCP sessions suffered a 6.5% cell-loss, which in turn caused a 38% retransmission in TCP. Traditional packet-switched networks will drop entire data packets during network congestion. The dropped packets will no longer consume bandwidth, buffer, and processing resources on the network. Later on, the sending workstation needs to retransmit these packets after their acknowledgments have timed out. While losing packets on a traditional network causes performance degradation, they are not as costly as the penalties incurred by cell-drops in congested ATM networks. The ATM technology segmentes data packets into many 53-byte ATM cells. When the congested ATM switch drops one or more ATM cells of a data packet, the entire data packet is corrupted and has to be retransmitted. Further, lost cells can belong to many different data packets; therefore a small cell-loss rate usually translates into a large TCP retransmission rate. In addition, ATM switches do not have the concept of data-packet; therefore, they will indiscriminately forward all cells, regardless of whether these cells belong to good or corrupted TCP packets. Forwarding cells from corrupted TCP packets wastes network bandwidth, switch buffers, and processing power at the destination end-station to perform cell reassembly.

The average round-trip-time estimated by TCP is presented in Figure 6. As shown by the graph, the TCP RTTs are significantly increased in the flow controlled case. TCP calculates its RTTs based on packets which are successfully delivered and does not include delays introduced by packet time-outs. For this reason, even though the RTT values for the simple cell-drop scheme appears smaller than that of the flow-controlled values, the delay introduced by TCP retransmissions and by the slow-start process are, in fact,much worse. Under congestion, the AN2 flow control scheme back-pressures the VCs along the communication path in order to prevent cell-loss. As a result, traffic is held at the edge of the network and the TCP RTTs are increased.

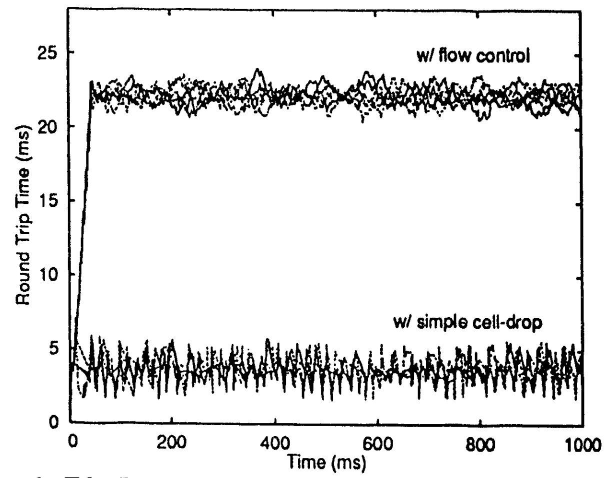


Figure 6. TCP Round Trip Time with flow control and simple cell-drop

Degree of Network Congestion

The degree of network congestion increases with the number of concurrent TCP sessions in Figure 2. We also studied the the effect of increased network congestion on TCP performance. Figure 7 plots the the average flow-controlled throughput and RTT, as well as the average throughput for the simple cell-drop scheme. As shown in Figure 7, the flow-controlled RTTs scale linearly with the degree of congestion. As demonstrated by results in Table 2, the increase in TCP RTTs only affects bandwidth-intensive applications such as large file transfers. These applications are usually insensitive to delay and can achieve high throughput performance using well-tuned TCP window. Besides, link level flow control makes a network more predictable which will help TCP in making RTT estimations. The ability for TCP to accurately estimate its RTT will help eliminate premature TCP retransmissions and improve TCP performance. Again, the AN2 flow control achieves full link utilization. The flow-controlled throughput curve in Figure 7 represents a plot of (link bandwidth/N) against N, where N is the number of contending sessions. The performance degradation of the simple cell-drop scheme can be clearly seen from the figure.

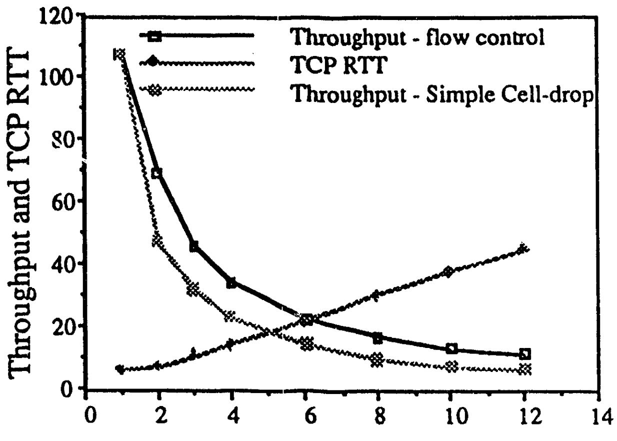


Figure 7. The effect of Network Congestion on TCP Throughput and RTT.

Effective Link Utilization

Figure 8 plots the percent utilization of the inter-switch link in Figure 2, for the AN2 flow control and the simple cell-drop scheme. Results show that with flow control the effective link utilization rises sharply and approaches the steady state value of 100%. With simple cell-drop, the raw link utilization rises slowly and reaches a lower steady-state value, and its effective link utilization levels off at a steady-state value of 63%. When cells are lost due to network congestion, TCP packets will be corrupted and, as a result, the corrupted packets will be discarded by the destination node. Acknowledgment time-outs will activate TCP's congestion avoidance scheme, the slow start algorithm, which will reduce TCP's window to one maximum transfer unit (MTU); the window recovery is gradual [10]. When all six sessions are periodically undergoing slow start cycles, the link can be under-utilized.

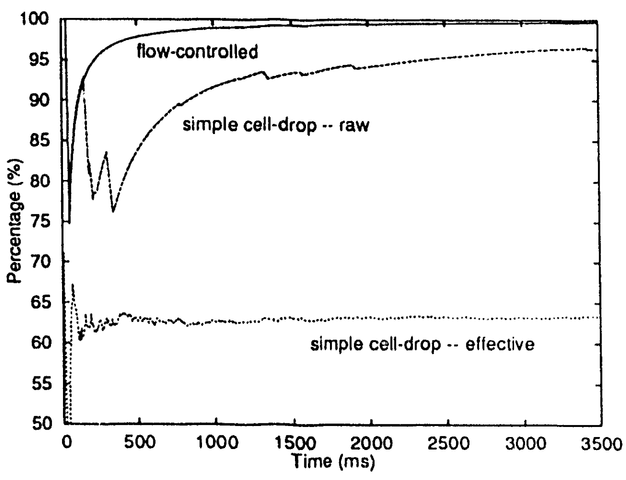


Figure 8. Link Utilization with flow control and simple cell-drop

As shown in Figure 8, the difference between the raw link utilization and the effective link utilization of the non-flow controlled runs is approximately 32% at steady state. Therefore, roughly 117000 useless cells (30% of OC-3 bandwidth) are being processed per second in an already congested ATM network. This is not only a big waste in "link bandwidth", but also in "switch buffer and processing" and "destination buffer and processing".

Selective Cell-drop Schemes

As shown in the previous section, without ATM-level flow control, a large percentage of the link bandwidth can be wasted transfering useless cells. If the wasted bandwidth can be used to transport useful cells, significant performance improvement can be realized. To verify this observation, we studied selective cell-drop as a form of congestion control. Selective cell-drop schemes require intelligent ATM switches which will drop cells belonging to the same corrupted TCP packet. We simulated two selective cell-drop schemes for our study. Both schemes require buffering on a per VC basis. The first scheme we simulated is the drop-tail scheme. In this scheme, whenever a per VC queue is full, an incoming cell to the queue is dropped. Subsequent incoming cells to this queue will be dropped until an End-of-Message (EOM) cell arrives; the EOM cell will not be dropped because it is the TCP packet delimiter in AAL5. Therefore, the drop-tail scheme will drop the tail of any IP packet starting from the first cell in the packet which encounters congestion. The advantage of this scheme is that it is relatively easy to implement. However, this scheme does not discard all the cells belonging to the IP packet whose tail was dropped.

The second scheme we simulated is the drop-whole scheme. In this scheme, in addition to dropping the tail of the IP packet, as in the drop-tail scheme, the head of the packet is also dropped. Whenever an incoming cell is dropped as a result of queue overflow, cells from the end of the per VC queue are dequeued and dropped, up to but not including the EOM cell of the previous packet. Then, a similar algorithm as the drop-tail scheme is employed to drop subsequent cells until the end of the packet, including the EOM cell. The advantage of this scheme is that all cells belonging to a corrupted TCP packet will be dropped, leading to more efficient utilization of link bandwidth.

Figures 9, 10 and 11 show our simulation results of the two cell-drop schemes for the same network configuration as in Figure 2. Figure 9 compares the throughput performance of the TCP sessions for the two schemes. It can be observed that the drop-tail scheme results in an average TCP throughput of 18.5 Mbps, representing a 20% decrease in throughput over that of the flow-controlled scheme and a 24% improvement over that of the simple cell-drop sessions. Figure 9 also shows that the drop-whole scheme results in a more significant improvement in throughputs. The average throughput for the six TCP sessions is approximately 22.5 Mbps, representing a 3% decrease in throughput over that of the flow controlled scheme and a 50% improvement over that of the performance of simple cell-drop scheme.

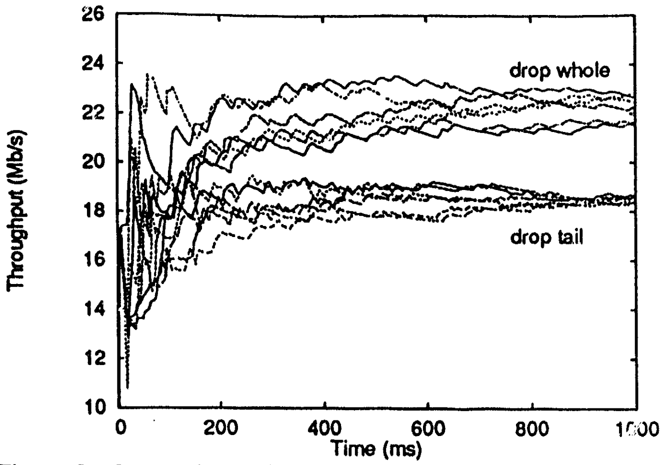


Figure 9. Comparison of the TCP throughput performance of the two selective cell-drop schemes.

Figure 10 and 11 shows the cell loss and TCP retransmission percentage for the two cell-drop schemes. Figure 10 shows that the drop-tail scheme results in an average of about 25% TCP retransmission, compared with 38% for the simple cell-crop scheme. Figure 11 shows that by freeing up additional bandwidth, the drop-whole scheme is able to reduce the TCP retransmission percentage to about 18%. Furthermore, with the drop-whole scheme the cell-drop rate is roughly equal to the TCP retransmission rate. This observation indicates that in the drop-whole scheme all of the useless ATM cells are dropped and that the raw link utilization is the effective link utilization. A comparison of Figures 4, 10, & 11 shows that the simple cell-drop scheme has the lowest cell-loss rate, 6.5%, but the highest TCP retransmission rate, 38%. On the other hand, the drop-whole scheme has the highest cell-drop rate, 17%, but the lowest TCP retransmission rate, 18%. The drop-tail scheme falls in the middle with a 7.5%cell-loss rate and a 25% TCP retransmission rate. These values illustrate that an effective discard of useless cells can improve effectiveness link utilization.

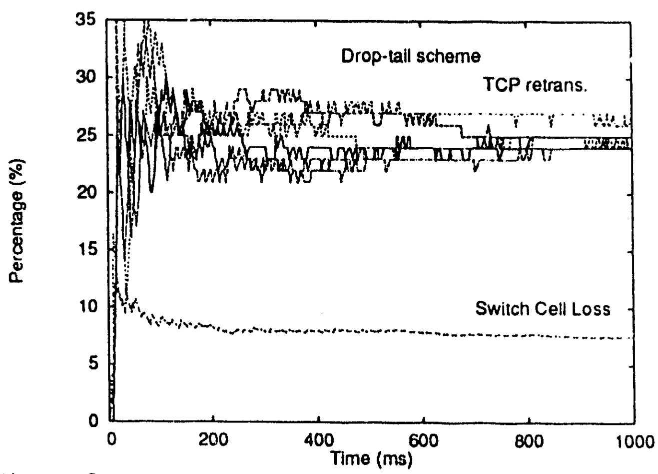


Figure 10 The cell-loss and TCP retransmission rates for the drop-tail scheme.

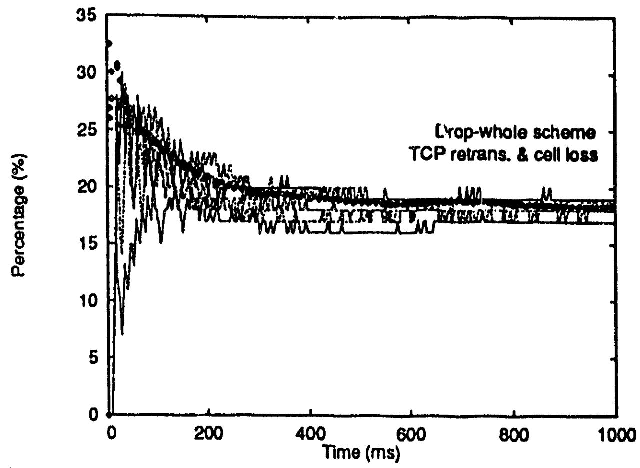


Figure 11. The cell-loss and TCP retransmission rates for the drop-whole scheme. The curve with Symbols represents cell loss.

Effect of Switch Buffer Size on TCP Performance

In previous simulation results, a buffer size of 256 is used for all cell-drop schemes. In this section we will study the effect of per VC buffer size on TCP performance for the various cell-drop congestion control schemes. Figures 12(a) and 12(b) plot the throughputs and TCP retransmission rates for the three cell-drop schemes versus buffer size. As expected, larger buffer size has a positive impact on TCP performance, especially for the drop-tail and simple cell-drop schemes. Note that when the per VC buffer is large enough to contain data in a maximum TCP-window of 64 KBytes (1368 cells), the TCP sessions achieved zero cell-loss and zero TCP retransmission for all cases. The corresponding throughputs become identical to the flow controlled throughput of 23.2 Mbps. Although large buffers can prevent cell-loss and achieves comparable TCP performance as the AN2 flow control scheme, the large buffer requirement tends to be prohibitively expensive, especially over a wide-area network. Since the drop-tail and the simple cell-drop schemes can achieve better performance only through large buffers, they are unlikely to be viable to provide congestion control in ATM networks.

However, the drop-whole scheme can achieve high throughput even at the per VC queue size of 256 cells (Figure 12). As shown, at per VC queue of 256 cells, the throughput value (22.5 Mbps) of the drop-whole scheme is already at 98% of the optimal value (23.2 Mbps). The high throughput performance of the drop-whole scheme can be contributed to TCP's effective reactive congestion control mechanism for packet networks; the drop-whole scheme resembles a packet network in that it discards cells of an entire IP packet during network congestion. Even though its TCP throughput performance is very good, the drop-whole scheme has a 17% TCP retransmission rate at the per VC queue size of 256 cells. This retransmission rate can introduce excessive delay and can be a concern to interactive and real-time applications, especially when the TCP retransmission timer granularity is at 500 ms with the slow-timer mechanism. At per VC queue size of 512 cells, however, the TCP retransmission rate is reduced to around 7.5%, representing a marked increase in performance. But even 256 cells per VC is a large memory requirement. Given the high performance of the drop-whole scheme, a cost-performance comparison study with the AN2 flow control scheme can be interesting.

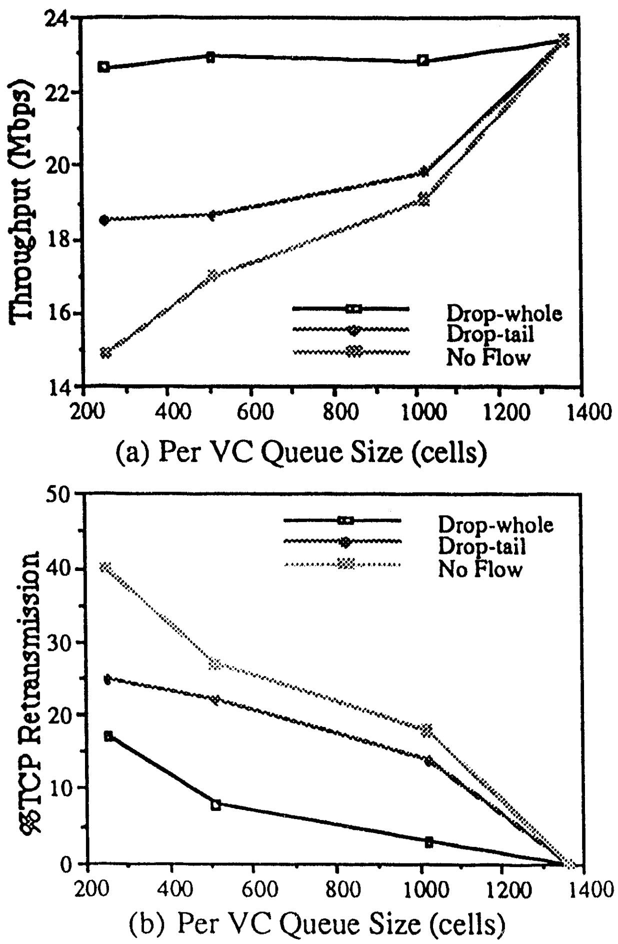


Figure 12. Effect of Per VC Queue Sizes on (a) TCP Throughput and (b) Retransmission

Conclusion and Future Work

We have presented a simulation study of the performance of TCP in a congested ATM local area network. Our simulation results show that with a simple cell-drop scheme TCP suffered severe performance degradation during congestion. Selective cell-drop schemes can improve TCP performance by reclaiming some of the wasted bandwidth in a simple cell-drop scheme. However, zero cell loss, full link utilization, and maximal TCP performance are achieved with the ATM-level AN2 flow control scheme.

A comparison of TCP performance for the various proposed ATM-level congestion control schemes [15, 20, 18, 12] would be of great interest to the networking community, as they would reveal interesting interaction between the ATM layer and higher-layer protocols. An ongoing effort at Sandia is to include in our simulation model the above proposals as well as other transport-layer protocols, such as IPX, NFS etc., and different source traffic models. These additions will provide a more comprehensive set of metrics for evaluating the various congestion control proposals.

References

[1] T. E. Anderson and S. S. Owicki and J. B.Saxe and C. P. Thacker, High Speed Switch Scheduling for Local Area Network, DEC Internal Publication,1993.

[2] G. J. Armitage and K. M. Adams, Packet Reassembly During Cell Loss, IEEE Network Magazine, Vol. 7, No. 5, pp. 26-34, September 1993.

[3] G. J. Armitage and K. M. Adams, Some Observations on the ATM Adaption Layer Services Needed to Support TCP/IPS, Proc. Australian Broadband Switching and Services Symposium '92, Vol. 2, July 1992.

[4] H. Y.Chen and J. A. Hutchins and N. Testi, TCP Performance over Wide Area Networks, Sandia National Laboratories Internal Report SAND93-8243, September 1993.

[5] G. Clapp and M. Zeug, Components of OSI: Asynchronous Transfer Mode (ATM) and ATM Adaptation Layers, Connexions: The Interoperability Report, Vol.6, No. 4, pp. 22-28, April 1992.

[6] D. Comer, Internetworking with TCP/IP: Principles, Protocols, and Architecture, pp. 129-150, Prentice Hall, 1988.

[7] A. E. Eckberg, B. T. Doshi and R. Zoccolillo, Controlling Congestion in B-ISDN/ATM: Issues and Strategies, IEEE Comm. Magazine, Vol. 29, No. 9, pp. 64-70, Sept. 1991.

[8] ATM Forum, ATM User-Network Interface Specification Version 3.0, Sept. 1993

[9] I. W. Habib and T. N. Saadawi, Controlling Flow and Avoiding Congestion in Broadband Networks, IEEE Communications Magazine, Vol. 29, No. 10, pp. 46-53, Oct. 1991.

[10] V. Jacobson, Congestion Avoidance and Control, Proc. ACM SIGCOMM'88, pp. 314-329, August 1988.

[11] A. Kolarov and G. Ramamurthy and C. Ikeda and H. Suzuki, Evaluation of Congestion Control Schemes for ABR Service, NEC USA, January, 1994 (NEC internal publication, request by email kolarov@ccrl.nec.nj.com).

[12] H. T. Kung and A. Chapman, The FCVC (Flow Controlled Virtual Channels) Proposal for ATM Networks, Jan. 1994 (available via anonymous ftp from virtual.harvard.edu:/pub/htk/atm-forum/fcvc.ps).

[13] H. T. Kung and R. Morris, T. Charuhas and D. Lin, Use of Link-by-Link Flow Control in Maximizing ATM Networks Performance: Simulation Results, Proc. IEEE Hot Interconnects Symposium '93, Palo Alto, CA, Aug. 1993.

[14] D. Martin, Network Simulator User's Manual, (ananymous ftp from spice.lcs.mit.edu:/pub/netsim), MIT, September 22, 1988.

[15] P. Newman, Backward Explicit Congestion Notification for ATM Local Area Networks, Proc. IEEE GLOBECOM '93, pp. 719-723, December 1993.

[16] M. de Prycker, Asynchronous Transfer Mode - solution for broadband ISDN, pp. 97-124, Ellis Horwood, 1991.

[17] A. Romanow and S. Floyd, "Dynamics of TCP Traffic over ATM Networks", in Proc. SIGCOMM '94, Aug. 1994

[18] J. Scott and B. Simcoe, Digital Flow Control, ATM Forum Contribition 93-778, July 1993.

[19] W. Stallings, Networking Standards: A Guide to OSI, ISDN, LAN, and MAN Standards, Addision-Wesley Publishing Company, pp. 364-397, 1993.

[20] N. Yin, and M. Hluchyi, On Closed-Loop Rate Controls for ATM Cell Relay Networks, to be presented at IEEE INFOCOM '94, Sept. 1994.

DATE FILMED

10/18/94

END