Dynamics of TCP Traffic over ATM Networks

Allyn Romanow* and Sally Floyd†
allyn@eng.sun.com, floyd@ee.lbl.gov

This is a slightly-expanded version of a paper that appeared in
IEEE Journal on Selected Areas In Communications, May, 1995.

Abstract

We investigate the performance of TCP connections over ATM networks without ATM-level congestion control, and compare it to the performance of TCP over packet-based networks. For simulations of congested networks, the effective throughput of TCP over ATM can be quite low when cells are dropped at the congested ATM switch. The low throughput is due to wasted bandwidth as the congested link transmits cells from 'corrupted' packets, i.e., packets in which at least one cell is dropped by the switch. We investigate two packet discard strategies which alleviate the effects of fragmentation. Partial Packet Discard, in which remaining cells are discarded after one cell has been dropped from a packet, improves throughput somewhat. We introduce Early Packet Discard, a strategy that prevents fragmentation and brings throughput up to maximal levels by having the switch drop whole packets prior to buffer overflow.

1 Introduction

We investigate basic questions of congestion control for best-effort traffic in ATM networks. ATM (Asynchronous Transfer Mode) is the 53-byte cell-based transport method chosen by ITU-TSS (formerly CCITT) for Broadband ISDN [D93]. ATM is also under development for use in high speed LANs. In the ATM context, best-effort traffic is data traffic that does not have stringent real-time requirements. It is called Available Bit Rate (ABR) traffic in the ATM Forum.[1] Despite its importance, few studies prior to this one have investigated the performance of ATM for best-effort data traffic (exceptions are [C91, SC93, C93, HKM93]).

Since ATM does not provide Media Access Control, it has been a concern that the throughput will be low if an ATM network experiences congestion; in fact there is already practical evidence to this effect [C94]. This paper uses simulation studies to investigate the throughput behavior of TCP over ATM for best-effort traffic when there is network congestion. First, we consider the throughput performance of TCP over ATM without any additional ATM/AAL-level[2] congestion control mechanisms, such as hop-by-hop flow control or ATM-level rate-based traffic management. We compare this TCP over *plain* ATM with the performance of *Packet TCP*, TCP that is not over ATM and therefore is not fragmented into cells by the link layer.

The simulation results comparing TCP over plain ATM with Packet TCP show that the former can have arbitrarily low throughput performance, whereas Packet TCP has quite good performance. We examine these results and explain the dynamics causing the poor behavior. Based on this analysis, we explore two simple control mechanisms: Partial Packet Discard and Early Packet Discard. In Partial Packet Discard [AA93], if a cell is dropped from a switch buffer, the subsequent cells in the higher layer protocol data unit are discarded. We find that Partial Packet Discard improves performance to a certain degree, but that throughput performance is still not optimal. To improve performance further, we propose a mechanism called Early Packet Discard that brings throughput performance to its optimal level. In Early Packet Discard when the switch buffer queues reach a threshold level, entire higher level data units (e.g., TCP/IP packets) are dropped.

This study is focused on the dynamics of TCP (Transport Control Protocol) over ATM networks.[3] The performance of TCP is important as the protocol is widely used in today's Internet and private networks [S94]. Although this study is specific to TCP, the results are relevant to any packet-based protocol running over ATM. Thus the general results apply to UDP (User Datagram Protocol), or to other transport protocols such as DECnet [JR88], IPX, VMTP [C88], and AppleTalk [SAO90].

Analysis of throughput behavior shows that the poor perfor-

*Sun Microsystems Inc., 2550 Garcia Ave., Mountain View, CA 94043.
†Lawrence Berkeley Laboratory, 1 Cyclotron Road, Berkeley, CA 94720.
This work was supported by the Director, Office of Energy Research, Scientific Computing Staff, of the U.S. Department of Energy under Contract No. DE-AC03-76SF00098.

[1] The ATM Forum is an international consortium whose goal is to accelerate the use of ATM products and services through the development of interoperability specifications and the promotion of industry cooperation.

[2] The ATM Adaptation Layer (AAL) is the layer above the ATM layer that adapts the fixed-size ATM cell to the next higher-level data unit, which may be either signaling or user data [CCITT362, D93].

[3] For the purposes of this paper, IP is relevant only in that it determines the size of the higher level packet.

mance of TCP over plain ATM is caused by the well-known problem of fragmentation. In this case, TCP/IP packets are fragmented at the ATM layer. When the ATM switch drops a cell, the rest of the higher-level packet is still transmitted, clogging the congested link with useless data. Thus some additional control mechanism is required to achieve acceptable performance for TCP in ATM networks. Our simulations also show that for plain ATM, smaller switch buffers, larger TCP windows, and larger TCP packet sizes can each reduce the overall effective throughput.

This paper focuses on the specific traffic goal of achieving high throughput for TCP over ATM. Several additional goals of traffic management for best-effort traffic not under consideration here include fairness, low average delay and the control of misbehaving users. We assume the network context of fully supported Quality of Service, in which best-effort traffic is one of several traffic classes, each of which is handled according to its respective service requirements.

Section 2 describes the experimental design. Sections 3, 4, and 5 discuss the simulation results for TCP over plain ATM, for Partial Packet Discard, and for Early Packet Discard. Section 6 considers the use of additional congestion control mechanisms along with Early Packet Discard, and Section 7 discusses future work.

2 Simulation Set-up

This section describes the simulation environment. We compare simulations of TCP traffic over ATM networks with simulations of Packet TCP. The simulations use the network topology shown in Figure 1.

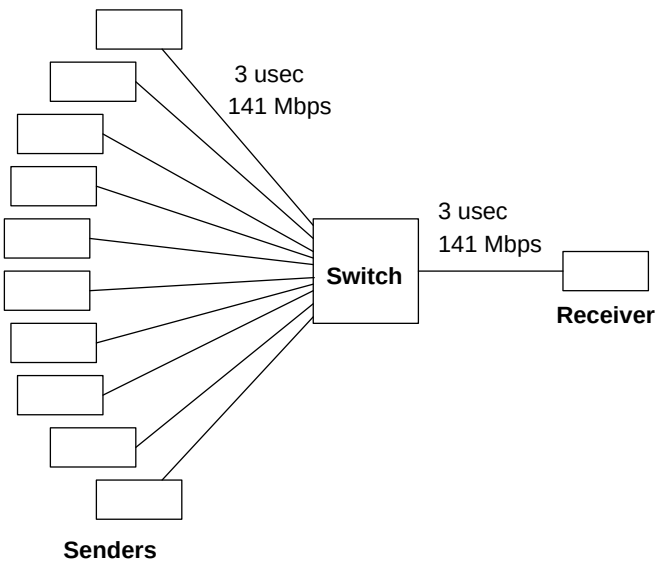


Figure 1: Simulation Scenario

A simple topology was chosen to make it easier to understand performance dynamics. Because a LAN environment has fewer switches and connections have a shorter roundtrip time, the congestion control issues are more straightforward than in a wide-area environment. To model a LAN environment, we used a propagation delay of 3 μsec for each link. Note that in this setting, the delay-bandwidth product is only four cells.

Simulations were run with ten "bulk-data" TCP connections. The number 10 was chosen to represent a relatively large number of simultaneous TCPs contending for the same resources. Each TCP connection is assumed to have an infinite supply of data. The simulation time was 15 seconds, which is reasonably long, relative to the average roundtrip time, and allows an aggregate transfer of more than 250 MB of data. In both the packet switch and ATM switch, the output ports use FIFO queuing.

All simulations were run for a range of values for the TCP packet size, switch buffer size, and TCP window size. Packet sizes were chosen according to common link level maximum transfer units (MTUs). The TCP packet size often used in IP networks is 512 bytes; 1500 bytes is the packet size for ethernet networks; 4352 bytes is for FDDI networks; and 9180 is the default for IP over ATM [A93]. Buffer size per output port ranges from 256 cells to 8000 cells. A buffer size of 1000 or 2000 cells per port for a small switch (e.g., 16 - 32 ports) would not be considered unusual. TCP window size uses common values of 8 kB, 16 kB, 32 kB, and 64 kB.

In the simulations of TCP over ATM, each TCP connection has its own Virtual Channel (VC), identified by a Virtual Channel Identifier (VCI) [CCITT150]. The link bandwidth was set to 141.333 Mbps, so that the cell transmission time for a 53-byte cell is an integral number of microseconds (3 μsec). The ATM simulator is a modified version of one developed by Lixia Zhang. The simulations for Packet TCP use a modified version of the REAL simulator [K88], with extensive enhancements by Steven McCanne and Sugih Jamin [FJ92].

Both simulators implement the same version of TCP based on the 4.3-Tahoe BSD release [J88]. Briefly, there are two phases to TCP's window-adjustment algorithm. A threshold is set initially to half the receiver's advertised window. The connection begins in slow-start phase, and the current window is essentially doubled each roundtrip time until the window reaches the threshold. Then the congestion-avoidance phase is entered, and the current window is increased by roughly one packet each roundtrip time. The window is never allowed to increase to more than the receiver's advertised window.

In 4.3-Tahoe BSD TCP, packet loss (a dropped packet) is treated as a "congestion experienced" signal. The source uses the *fast retransmit* procedure to discover a packet loss: if four ACK packets are received acknowledging the same data packet, the source decides that a packet has been dropped [S94]. The source reacts to a packet loss by setting the threshold to half the current window, decreasing the current window to one, and entering the slow-start phase. The source also uses retransmission timers to detect lost packets.

To improve the performance of TCP in the high-speed low-propagation-delay environment modeled in these simulations, we adjusted some parameters in TCP's algorithm for setting the retransmit timeout values. The current TCP algorithms for

measuring the roundtrip time and for computing the retransmit timeout are coupled [J88] and assume a coarse-grained clock. We changed the clock granularity for measuring the roundtrip time to 0.1 ms, and we changed the initial values used to set the retransmit timeout. These changes were necessary in order for the simulation results not be dominated by the specifics of current implementations of TCP. Our goal is to explore the possible limitations that ATM networks place on the performance of TCP-style protocols, and this must be distinguished from artifacts of specific implementation choices. TCP's retransmit timer algorithms are discussed in more detail in Appendix B.

The simulated ATM switch is based on a design by Bryan Lyles [LSB94]. The switch modeled is a 16-port output-buffered Batcher-banyan design. The switch architecture is of a sufficiently general design that the performance results are not architecturally dependent. In particular, the simulation results are not dependent on whether the switch is input-buffered or output-buffered. In the simulated ATM switch, the input cell queues are served in round robin fashion. In one cell time, a cell from each of two input port queues can be queued onto the same output port. If the output buffer is full, then only one of the two cells can be queued and the other cell must be dropped. In this case, each of the two contending cells has equal probability of being dropped.

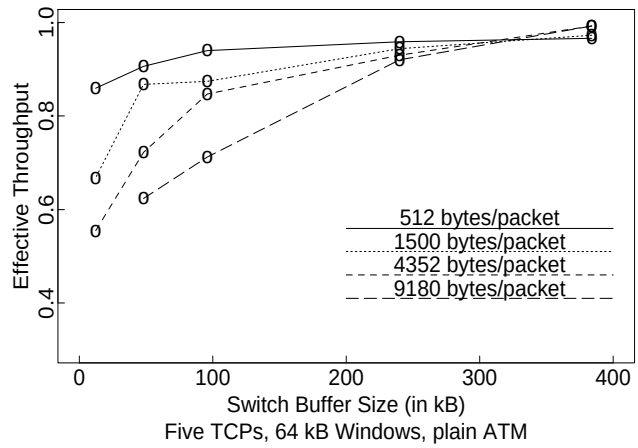
For the simulations with Packet TCP, the router has an output-buffered switch with a Drop-Tail packet-dropping discipline. That is, when the output buffer overflows, packets arriving at the output buffer are dropped.

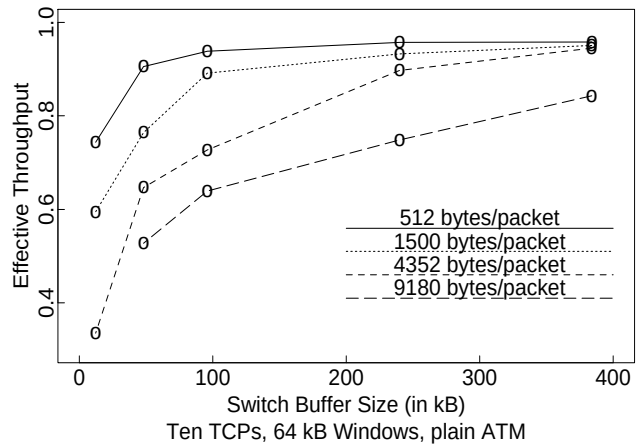
3 TCP over Plain ATM

It has been known for some time that packet fragmentation can result in wasted bandwidth and packet retransmission [KM87]. Because the network might drop only one fragment from a packet, "the loss of any one fragment means that the resources expended in sending the other fragments of that datagram are entirely wasted" [KM87, p.392]. In this section we quantify the effect of fragmentation with simulations of TCP traffic over plain ATM. Following sections discuss ways to avoid wasted bandwidth.

Figure 2 shows effective throughput as a function of switch buffer size and TCP packet size for TCP over plain ATM. We define the *effective throughput* or goodput as the throughput that is "good" in terms of the higher-layer protocol. That is, the effective throughput does not include cells that are part of a retransmission or an incomplete packet.[4]

The simulations are shown for both five and ten TCP connections. The top two graphs use a maximum TCP window of 64 kB, and the bottom graph uses a maximum TCP window of 8 kB. The y-axis shows the effective throughput as a fraction of the maximum possible effective throughput. On the x-axis is the switch output buffer size in kilobytes (ranging from 256 to 8000 cells). Circles are shown in the plots at 256, 1000, 2000, 5000, and 8000 cells. Different line types represent simulations with different packet sizes.[5]





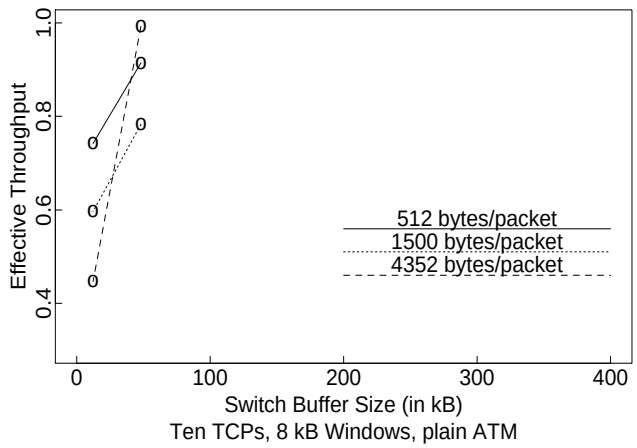


Figure 2: TCP over Plain ATM

Figure 3 shows the simulation results for ten TCP connections with Packet TCP, for the same range of buffer sizes and packet

[4]For a packet that does not break down into an integer number of cells, the effective throughput does not include bytes from the "padding" in the last cell. See [C92] for proposals to reduce bandwidth inefficiencies due to such size mismatch.

[5]For the case of a 256-cell buffer and packet size of 9180 bytes, the data point is not shown because results were segregated. Segregation is discussed in more detail in Appendix A.

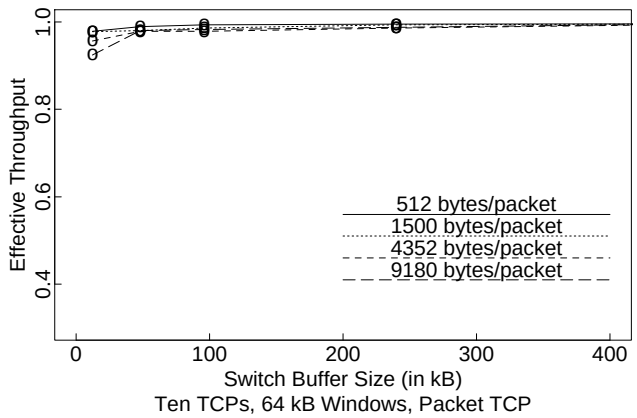


Figure 3: Packet TCP over Non-ATM Networks

sizes used in Figure 2. Comparing the two figures, it is clear that TCP over plain ATM can have lower throughput than Packet TCP. For the simulations with Packet TCP, the effective throughput is always at least 90% of the maximum possible, while for simulations of TCP over plain ATM the effective throughput can be as low as 34%.[6]

Figure 2 shows that smaller switch buffers and larger TCP packet sizes both reduce the effective throughput for TCP over plain ATM. The TCP window size also affects the effective throughput of TCP over plain ATM. Simulations with the TCP window ranging from 8 kB to 64 kB show a lower effective throughput with larger windows. Simulations of TCP over plain ATM with both five and ten TCP connections show that the effective throughput is lower with increased congestion caused by a larger number of TCP connections.

As Figure 3 shows, the effective throughput for Packet TCP is not greatly affected by either buffer size or packet size. Similarly, aggregate effective throughput is high even with many TCP connections or large TCP windows.

The following section explains these effects in detail.

3.1 Analysis

In this section we consider three possible causes for low effective throughput for TCP over ATM. These are the delivery of inactive cells, link idle time, and the retransmission of packets that have already been received.

The primary reason for the low effective throughput of TCP over ATM in our simulations is that when cells are dropped at the switch, the congested link transmits other cells from "corrupted" packets (that is, packets with at least one cell dropped by the switch). In the absence of fragmentation, this phenomenon does not occur in packet TCP, where packets dropped at the switch are not transmitted over the congested link. This problem of lost throughput due to "dead" cells transmitted on the congested link 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.

Larger packet sizes increase the number of wasted cells that the congested link transmits when the switch drops a single cell from one packet. In addition, the use of larger TCP packets substantially increases the aggressiveness of TCP's window increase algorithm, which in the congestion avoidance phase increases the congestion window by roughly one packet per roundtrip time. While larger packet sizes may be considered advantageous because they do not fragment Network File System (NFS) packets (which default to 8 kB), and because some end-nodes can process larger packets more cheaply than smaller packets [A93], large packets are a performance disadvantage in a congested local-area ATM network.

A secondary reason for the low effective throughput in some of the simulations is that the congested link is occasionally idle. However, except for a few simulations with very small buffers, where link idle time reached 20% of the link bandwidth, the amount of link idle time was typically close to zero. For the scenarios explored in this paper, this global synchronization is not a significant problem; connections recover fairly quickly from a dropped packet, and a single connection with a small window is sufficient to keep the congested link highly utilized.

Despite the fact that link idle time is not an important factor in these simulations, it could be a significant problem with TCP over plain ATM [KMP94]. The phenomenon of link idle time due to synchronization of the TCP window adjustment algorithms has been studied for Packet TCP [SZC90]. This synchronization could be exacerbated in ATM networks, where cells from different packets are usually interleaved at the switch, causing several TCP connections to synchronize and go through slow-start at roughly the same time. Link idle time can be affected by details of the retransmit timer algorithms (discussed in the Appendix).

A third possible reason for low effective throughput, but one that is not a significant factor in our simulations, is that the congested link could retransmit packets that have already been correctly received. With current TCP window adjustment algorithms, this can only occur when multiple packets are dropped from one window of packets. In these ATM LAN simulations, due to the congestion, windows generally do not become sufficiently large for this to be a significant problem.

The importance of network configuration parameters - switch buffer size, TCP packet size, and TCP window size - should not suggest that the fragmentation problem can be completely solved by appropriate configuration settings, which offer only partial solutions. Large buffers can result in unacceptably long delay, and it is not always possible to use small packets in an internet environment. In addition, the beneficial effect of small windows, small packets, or large buffers can be offset if the number of contending connections increases.

[6] Although the simulations in Figures 2 and 3 were run on different simulators, we have made substantial efforts to reduce secondary effects in the simulations that could obscure the main dynamics. This is described in more detail in the appendices.

4 ATM with Partial Packet Discard

Since the main problem with TCP over plain ATM is that useless cells congest the link, an alternative strategy is for the ATM switch to drop all subsequent cells from a packet as soon as one cell has been dropped. We refer to this strategy as *Partial Packet Discard* (PPD). It was called "selective cell discarding" in [AA93]. A similar cell-dropping mechanism was developed in the Fairisle ATM switch [LM91, p. 330]. In a different context, [RRV93] proposed dropping subsequent packets of video frames after the loss of a subset of packets invalidates the entire frame.

Implementing PPD would be quite straightforward with AAL5 [AA93], an adaptation layer designed for data and standardized by ITU-TSS [CCITT363, ANSI91]. PPD could be signaled on a per-VC basis. With Partial Packet Discard, once the switch drops a cell from a VC, the switch continues dropping cells from the same VC until the switch sees the ATM-layer-user-to-user (AUU) parameter set in the ATM cell header, indicating the end of the AAL packet. The end-of-packet cell (EOP) itself is not dropped. Because AAL5 does not support the simultaneous multiplexing of packets on a single VC, the AUU parameter can be used to delimit packet boundaries.

The implementation of Partial Packet Discard requires the switch to keep additional per-VC state in order to recognize which VCs are using AAL5 and want to use PPD; this can be established through ATM-level signaling. Also, the switch must keep state on which VCs are currently having cells dropped. A related approach that would not require using the AUU parameter would be for the switch to discard a significant number of cells from a single VC when the switch is forced to drop cells. The performance of this approach should be somewhat worse that the performance of Partial Packet Discard, because the switch could transmit dead cells from the first part of one corrupted packet, along with dead cells from the last part of another corrupted packet.

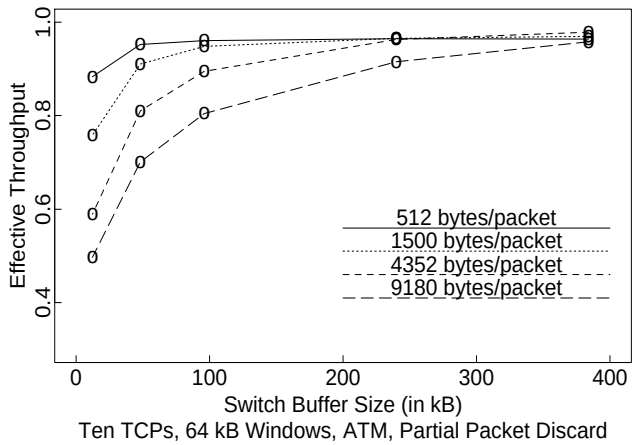


Figure 4: Partial Packet Discard

Figure 4 shows the throughput results for simulations with Partial Packet Discard, using the same configuration as in previous simulations. As Figure 4 shows, the effective throughput is improved with PPD. However, improvement is limited because the switch begins to drop cells only when the buffer overflows. The first cell dropped by the switch might belong to a packet that contains queued cells or cells already transmitted on the congested link.[7] Thus, the congested link can still transmit a significant fraction of cells belonging to corrupted packets.

5 Early Packet Discard

Effective throughput for TCP over ATM can be much improved if the switch drops entire packets prior to buffer overflow. This strategy, called *Early Packet Discard*, or EPD, prevents the congested link from transmitting useless cells and reduces the total number of corrupted packets. As with Partial Packet Discard, EPD can be signaled on a per-VC basis.

With Early Packet Discard, when a switch buffer becomes in danger of overflowing, the switch drops all of the cells in an AAL5 frame (i.e., it drops a packet). This is a violation of layering consistent with the growing trend towards Application Layer Framing and Integrated Layer Processing, which hold that "layering may not be the most effective modularity for implementation" [CT90, p. 200].

In the implementation of EPD for our simulation study, the switch drops packets whenever the proportion of the buffer in use exceeds a fixed threshold; in our simulations this threshold is set to half the buffer size. When the threshold is reached, the switch drops the first arriving cell and all subsequent cells of any incoming packet belonging to a VC designated as using EPD. As long as the buffer queue exceeds the fixed threshold, the switch continues to drop packets from multiple connections. In terms of the cell-dropping strategy, the ATM switch emulates a packet-based switch, dropping complete packets of cells.

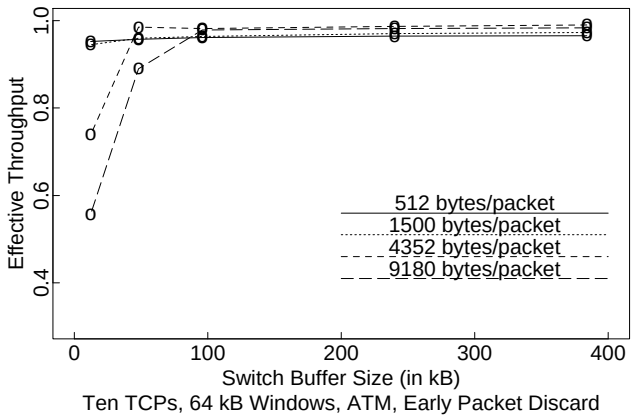


Figure 5: Early Packet Discard

Figure 5 shows the results using EPD in the same congestion

[7] In our implementation of Partial Packet Discard, when the switch first drops a cell, the switch does not look in the buffer for earlier cells that belong to the same packet, as such a search through the queue would be expensive to implement in hardware.

scenario previously investigated. Note that except for very small buffer sizes, high effective throughput is achieved.

The EPD mechanism does not require cooperation among ATM switches. It also does not rely on end-to-end congestion control to prevent uncontrolled buffer overflows. Thus higher level protocols such as UDP (User Datagram Protocol) that do not use end-to-end congestion control can benefit from EPD.

EPD affects congestion within a shorter time frame than do end-to-end congestion control mechanisms. EPD prevents useless data from being immediately transmitted. In contrast, TCP's end-to-end congestion control removes congestion in the longer term by reducing its congestion window in response to dropped packets.

In addition to keeping per-VC state, which is also required with Partial Packet Discard, the implementation of EPD requires the switch to monitor the active buffer queue size. The difficulty in supporting EPD depends on the details of a particular switch architecture; specifically, whether the functions of queue monitoring and packet dropping are located near each other in the hardware. In many switch designs these functions are co-located and it is relatively inexpensive to implement EPD.

TCP over ATM with EPD shares many of the dynamics of TCP in packet-switched networks. This includes a lack of protection from misbehaving users, global synchronization [ZC90], a bias against connections with longer roundtrip times [FJ92], and a bias against connections with multiple contested gateways [F91]. Early Packet Discard could have an additional bias against connections with shorter packets. If the congestion epoch is short and there are two active connections, one with small packets and one with large packets, the switch is likely to find the beginning of one of the smaller packets first, and might never need to drop cells from the connection with larger packets. These biases could be avoided by using additional congestion control mechanisms, as discussed in Section 6.

5.1 Preliminary Investigations on Setting the EPD Threshold

The goal of EPD is to prevent frequent buffer overflows by dropping complete packets before the buffer fills, rather than preventing *all* buffer overflows. Occasional buffer overflows do not have serious negative effects on performance.

The placement of the EPD threshold determines how efficiently the buffer is used, and how often cell dropping will occur. Setting the threshold depends on a great many factors and we have not made a complete study of the issue. This section offers some preliminary work.

We find it useful to distinguish two conceptually different aspects of buffer use. The EPD threshold essentially functions as the *effective buffer size*. The *excess buffer capacity*, referring to the buffer in excess of the EPD threshold, is used to accommodate cells from outstanding packets, those which have cells in transmission on the link. Consider an ATM network where the largest packet (i.e., AAL5 unit) contains m cells. If cells from different packets were never interleaved, then it would be sufficient to have an excess buffer capacity of m cells.

We show the relationship between excess buffer capacity and throughput for the restricted case which we have been using as an example - i.e., 64 kB windows, 10 TCPs, infinite traffic source model, fixed size packets. Figure 6 plots the effective throughput as a function of the excess buffer capacity, measured in numbers of packets. Here the EPD threshold, and thus the effective buffer size, is set to 1000 cells. The actual buffer size is set to 1000 cells plus a small multiple (from 1 to 5) of the packet size m. Different line types show the simulation results for different packet sizes. As Figure 6 shows, the effective throughput drops below 90% in those cases where the excess buffer capacity is less than $3m$ cells. Simulations with an effective buffer size of 2000 cells show similar results.

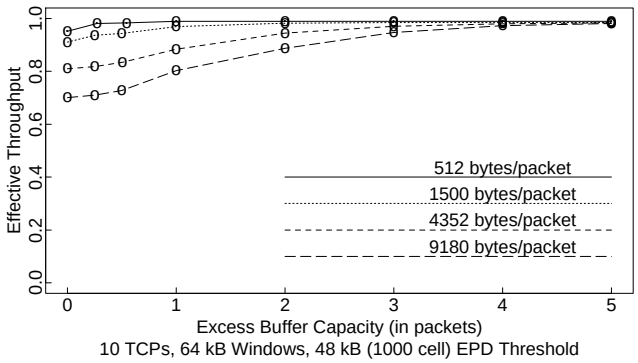


Figure 6: Effective Throughput vs. Excess Buffer Capacity

Further investigation is needed to determine the desired excess buffer capacity for more diverse and typical scenarios. A number of factors are relevant in determining the placement of the threshold which will lead to efficient operation. Among the most important considerations are the distribution of packet sizes and the traffic distribution. In addition, important factors include the duration of the congestion epoch; the proportion of the incoming cells during the congestion epoch that belong to outstanding packets, and thus need to be buffered; and the interaction with ATM-level or transport-level congestion control mechanisms. Also the amount of excess buffer capacity needed depends on how the buffer is shared with non-EPD traffic.

Determining the optimal value for the EPD threshold, that is, the effective buffer size, is a distinct issue from determining the optimal value for the excess buffer capacity. The question of the optimal effective buffer size depends primarily on the network context, e.g., LAN or WAN, tradeoffs between throughput and delay, the mix of non-EPD traffic, etc.

6 Speculations on Early Packet Discard with Other ATM Congestion Control Mechanisms

Although the EPD mechanism achieves high effective throughput for TCP or other best-effort traffic over ATM networks, it

does not address other traffic management concerns. Issues that may warrant further control mechanisms at the ATM level include controlling average delay for delay-sensitive best-effort traffic (e.g., telnet traffic), providing more stringent fairness requirements, protecting against misbehaving users, and reducing packet drops. This section considers how EPD might be used in conjunction with other ATM control mechanisms.

One possible way to achieve greater traffic control is to incorporate EPD with a RED gateway strategy. Random Early Detection (RED) gateway mechanisms [FJ93] are designed to maintain a low *average* queue size in cooperation with end-to-end congestion control mechanisms. If EPD and RED mechanisms were used together, the ATM switch would have two separate queue thresholds for two separate purposes. The EPD threshold, which would be fairly high, would be used so that the switch could drop complete packets of cells before being forced by a buffer overflow to drop partial packets of cells. The RED threshold, which would be lower, is a threshold for the *average* queue size. When the average queue size indicates persistent congestion, the switch would begin to inform sources to use their end-to-end flow control mechanisms to reduce the load on the network.

Two traffic management schemes for best-effort traffic are currently under development by the ATM Forum Traffic Management working group. One scheme is a feedback control scheme to operate at the ATM level, similar in nature to DECbit and TCP. In this scheme, the ATM-level source adjusts its sending rate in response to congestion indications from the ATM switches. Early versions appear in [M91, N93]. EPD could be used to make rate-based feedback control schemes more robust. By using EPD, feedback control schemes would be more tolerant of occasional cell loss, and could be designed to focus more heavily on other traffic management goals. This tolerance of occasional cell loss might be particularly useful in a switch designed to carry wide-area as well as local traffic.

The second proposal under investigation in the ATM Forum is a hop-by-hop credit-based strategy [KC94, S91]. This proposal is based on the goal of avoiding cell drops by allowing a VC to transmit cells on a link only when it is known that sufficient buffer capacity is available at the receiving switch. Given such a goal, EPD mechanisms would have little relevance. However, with the memory-sharing modifications that have been developed to adapt the hop-by-hop proposal to a wide-area environment [KBC94], problems could result if best-effort VCs receiving high bandwidth suddenly have their throughput reduced as a result of downstream congestion. In this case, the addition of EPD mechanisms to drop a packet of cells from stalled connections could free-up valuable memory, and at the same time provide early notification of congestion to the TCP source. Mechanisms similar to those in [FCH94] could be explored to implement EPD for ATM switches with per-VC buffers.

7 Conclusions and Future Work

As we have shown in this paper, TCP can perform poorly over plain ATM, primarily due to fragmentation. This situation is ameliorated by larger switch buffers and smaller sized TCP packets. Partial Packet Discard improves performance to some extent. However, the fragmentation problem can be obviated by Early Packet Discard, which is easily implemented with many switch designs.

For ATM networks with occasional transient congestion, PPD may be an adequate method to provide high effective throughput. Further investigations would be of interest of PPD and EPD strategies in different traffic environments, including a mix of packet sizes, diverse types of data, and different traffic flow models. It would also be interesting to study EPD and PPD with different ATM-level congestion-control mechanisms, such as those mentioned above. Also it would be useful to explore further the behavior of both discard mechanisms in an environment of multiple switches and a large number of VCs.

Of course, if ATM best-effort service is defined to require a cell-loss rate below 10^{-6}, then EPD is of little relevance. An important issue to consider is whether elimination of cell loss is desirable for best-effort traffic, which typically uses modern transport protocols such as TCP that have congestion control mechanisms at the transport layer. Currently such transport protocols rely on packet drops as the indication of congestion in the network. If packets are not dropped, the transport protocol will not be able exercise congestion control and may behave problematically. Further investigation of these issues is needed.

This work has pointed toward several relatively minor ways that TCP could be changed to accomodate native ATM. First, in order to allow high bandwidth utilization in a congested ATM LAN with high bandwidth and low propagation delay, we decreased the TCP clock granularity in our simulators by several orders of magnitude. Further investigation is needed on the effect of TCP clock granularity on overall network performance.

Another change in TCP that would accomodate ATM would be to dynamically vary the packet size. Smaller packet sizes could be used in LANs, where the advantages of larger packets are not as great, and where a less-aggressive window increase algorithm is appropriate.

A third change to TCP that would facilitate interactions with ATM-level congestion control mechanisms would be for TCP congestion control mechanisms to respond to explicit congestion notification, in addition to using packet drops as indications of congestion. We are currently exploring such modifications to TCP [F95].[8]

8 Acknowledgments

We thank Lixia Zhang for generous help with the ATM simulator and issues related to TCP dynamics. This work has also ben-

[8] For TCP sources that reduce their windows in response to explicit congestion notification, one requirement would be that a source would reduce its TCP window at most once per roundtrip time.

efited from discussions with Van Jacobson, Tom Lyon, Bryan Lyles, Peter Newman, Chris O'Neill, Vern Paxson, the Traffic Management Working Group of the ATM Forum, and others, and from feedback from anonymous reviewers. We thank Sugih Jamin for his useful comments on the paper, and Julie Haslam for her help.

References

[ANSI91] ANSI T1S1.5 91-449, "AAL5- A New High Speed Data Transfer AAL", ANSI T1S1.5, November 1991.

[A93] R. Atkinson, "Default IP MTU for Use Over ATM Adaptation Layer 5 (AAL5)", Working Draft, IETF draft-ietf-atm-mtu-06.txt, available by anonymous FTP from ds.internic.net:internet-drafts, June 1993.

[AA93] G. Armitage and K. Adams, "Packet Reassembly During Cell Loss", *IEEE Network Mag.*, Vol. 7 No. 5, Sept. 1993, pp. 26-34.

[CCITT150] CCITT SGXVIII - report R109, Draft recommendation I.150, "B-ISDN Asynchronous Transfer Mode Functional Characteristics", July 1992.

[CCITT362] CCITT SGXVIII Draft recommendation I.362, "B-ISDN ATM Adaptation Layer (AAL) functional description", June 1992.

[CCITT363] CCITT SGXVIII Draft recommendation I.363, "B-ISDN ATM Adaptation Layer (AAL) Specification", January 1993, Section 6.

[C91] R. Cáceres, "Efficiency of ATM Networks in Transporting Wide-Area Data Traffic", TR-91-043 International Computer Science Institute, Berkeley, CA, July 1991.

[C92] R. Cáceres, "Multiplexing Traffic at the Entrance to Wide-Area Networks", Ph.D. thesis, Report No. UCB/CSD 92/717, University of California at Berkeley, Dec. 1992.

[C93] R. Cáceres, "Multiplexing Data Traffic Over Wide-Area Cell Networks", unpublished technical report, Matsushita Information Technology Laboratory, Princeton, NJ, March 1993.

[C88] D. Cheriton, "VMTP: Versatile Message Transaction Protocol- Protocol Specification", RFC-1045, Feb. 1988.

[CT90] Clark, D. and Tennenhouse, D., "Architectural Considerations for a New Generation of Protocols", *Proc. SIGCOMM '90*, Sept. 1990, pp. 201-208.

[C94] M. Csenger, "Early ATM Users Lose Data", *Communications Week*, May 16, 1994.

[D93] M. DePrycker, *Asynchronous Transfer Mode*, Second Edition, Ellis Horwood, London, 1993.

[FCH94] C. Fang, H. Chen, and J. Hutchins, "Simulation Analysis of TCP Performance in Congested ATM LANs using DEC's Flow Control Scheme and two Selective Cell-Drop Schemes", January 1994, available by anonymous ftp from ca.sandia.gov:pub/papers/atm94-0119.ps. A longer paper including these results is to appear in Globecom 94.

[F91] Floyd, S., "Connections with Multiple Congested Gateways in Packet-Switched Networks Part 1: One-way Traffic", *Computer Communication Review*, V.21 N.5, October 1991, pp. 30-47.

[F95] Floyd, S., "TCP and Explicit Congestion Notification", to appear in *Comput. Commun. Rev.*, 1995.

[FJ92] S. Floyd and V. Jacobson, "On Traffic Phase Effects in Packet-Switched Gateways", *J. Internetworking: Research and Experience*, 3(3), Sept. 1992, pp. 115-156.

[FJ93] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance", *IEEE/ACM Trans. on Networking*, Vol. 1 No. 4, Aug. 1993, pp. 397-413.

[HKM93] E. Hahne, C. Kalmanek, and S. Morgan, "Dynamic Window Flow Control on a High-Speed Wide-Area Data Network", *Computer Networks and ISDN Systems*, Vol. 26 No. 1, Sept. 1993, pp. 29-41.

[J88] V. Jacobson, "Congestion Avoidance and Control", ACM SIGCOMM '88, Aug. 1988, pp. 314-329. (An updated version of this paper is available by anonymous FTP from ftp.ee.lbl.gov:congavoid.ps.Z.)

[JR88] R. Jain and K.K. Ramakrishnan, "Congestion Avoidance in Computer Networks with a Connectionless Network Layer: Concepts, Goals, and Methodology", *Proc. IEEE Comp. Networking Symp.*, Washington, D.C., pp. 134-143, Apr. 1988.

[KMP94] K. Keeton, B. Mah, and C. Paxson, "Characterizing ATM Performance: Weighing a Tomato on a Truck Scale", class report, UC Berkeley, May 1994.

[KM87] C. Kent and J. Mogul, "Fragmentation Considered Harmful", ACM SIGCOMM '87, Aug. 1987, pp. 390-401.

[K88] S. Keshav, "REAL: a Network Simulator", Report 88/472, Computer Science Department, University of California at Berkeley, Berkeley, CA, 1988.

[KC94] H. Kung and A. Chapman, "The FCVC (Flow-Controlled Virtual Channels) Proposal for ATM Networks", January 1994, available via anonymous FTP from virtual.harvard.edu:pub/htk/atm-forum/fcvc.ps.

[KBC94] H. Kung, T. Blackwell, and A. Chapman, "Adaptive Credit Allocation for Flow-Controlled VCs", March 1994, available via anonymous FTP from virtual.harvard.edu:pub/htk/atm-forum/ atm_forum_94-0282.ps.

[LM91] I. Leslie and D. McAuley, "Fairisle: An ATM Network for the Local Area", ACM SIGCOMM '91, Sept. 1991, pp. 327-336.

[LSB94] B. Lyles, D. Swinehart, and A. Bell, "Anatomy of an ATM Switch: The BADLAN Local Area ATM Design", (working title), in preparation, 1994.

[M91] B. Makrucki, "On the Performance of Submitting Excess Traffic to ATM Networks", IEEE GLOBECOM '91, Dec. 1991.

[N93] P. Newman, "Backward Explicit Congestion Notification for ATM Local Area Networks", *Proc. GLOBECOM '93*, Dec. 1993, pp. 719-723.

[RRV93] S. Ramanathan, P. Rangan, and H. Vin, "Frame-Induced Packet Discarding: An Efficient Strategy for Video Networking", *Fourth International Workshop on Network and Operating System Support for Digital Audio and Video*, Lancaster University, Nov. 93, pp. 175-186.

[SC93] A. Schmidt and R. Campbell, "Internet Protocol Traffic Analysis with Applications for ATM Switch Design", *ACM Computer Communication Review*, Vol. 23 No. 2, Apr. 1993, pp. 39-52.

[S91] M. Schroeder, A. Birrell, M. Burrows, H. Murray, R. Needham, T. Rodeheffer, E. Satterthwaite, and C. Thacker, "Autonet: A High-Speed, Self-Configuring Local Area Network Using Point-to-Point Links", *IEEE J. Select. Areas Commun.*, Vol. 9 No. 2, pp. 1318-1335.

[SZC90] S. Shenker, L. Zhang, and D. Clark, "Some Observations on the Dynamics of a Congestion Control Algorithm", *ACM Computer Communication Review*, Vol. 20 No. 4, Oct. 1990, pp. 30-39.

[SAO90] G. Sidhu, R. Andrews, and A. Oppenheimer, *Inside AppleTalk*, Addison-Wesley, Reading, MA, 1990.

[S94] W. Stevens, *TCP/IP Illustrated*, Volume 1, Addison-Wesley, Reading, MA, 1994.

[ZC90] Zhang, L., and Clark, D., "Oscillating Behavior of Network Traffic: A Case Study Simulation", *Internetworking: Research and Experience*, Vol. 1, 1990, pp. 101-112.

A Simulation Details

We used two different simulators for these experiments, one for the simulations of Packet TCP, and another for the simulations of TCP over ATM. Neither simulator is capable of running both kinds of simulations.

Occasionally in simulations one or more connections can be starved out (prevented from attaining any bandwidth by the other connections). This phenomenon, called segregation, has been explained in relation to phase effects in [FJ92]. In order to avoid these phase effects the simulations in this paper add a small random component to the roundtrip time for each packet [FJ92]. In addition, both the Packet TCP and the ATM simulations include an added telnet connection that uses less than 1% of the link bandwidth. The telnet connection further offsets phase effects by sending short packets at random times.

In the Packet TCP simulations, the link speed ranges from 141.176 Mbps to 141.528 Mbps, depending on the packet size. This gives a packet transmission time that is an integer number of microseconds, ranging from 29 μsec for a 512-byte packet to 520 μsec for a 9180-byte packet.

B TCP Retransmit Timers

The clock granularity for TCP can contribute to poor TCP performance in a high-speed low-propagation-delay environment with ATM. In TCP, the retransmission timer is set as a function of the roundtrip time. The minimum value for the retransmit timer is twice the TCP clock tick, where the TCP clock tick refers to the granularity of the clock used for measuring the TCP roundtrip time. If a retransmitted packet is itself dropped, an exponential backoff is triggered.

In the TCP Tahoe release, the clock granularity is typically 300-500 msec; this clock granularity can be too coarse for the high-speed low-propagation-delay ATM environment. With a clock granularity of 300 msec, when a TCP packet is dropped due to congestion, the retransmit timer gets set to a relatively large value, compared to the actual roundtrip time.

For the simulations reported here, we set the TCP clock granularity to .1 msec, which worked reasonably well for this environment.[9] Setting the clock to a finer granularity did not work well. Because the algorithms for setting the retransmit timer assume a somewhat coarse granularity for the TCP clock, relative to the measured roundtrip times, simulations with the clock granularity set to 1 μsec result in spurious retransmissions.

To further explore the effects of clock granularity, we ran simulations of Early Packet Discard with the TCP clock granularity set to 300 msec instead of 0.1 msec. For simulations with mild congestion, the effective throughout is still high, but there can be extreme unfairness, with several connections caught in exponential backoff while other connections receive most of the bandwidth. For simulations with severe congestion, a TCP clock granularity of 300 msec results in lower effective throughput.

[9] In addition, we adjusted time constants for initial values of the retransmit timer, so that if the first packet of a connection is dropped, the retransmit timer is 0.2 seconds. We staggered the start times of the various connections to reduce the probability of having the network drop the first packets of a connection, thereby reducing the impact of initial values of the retransmit timer.

However, the clock granularity may not be of critical importance in state-of-the-art and future TCP implementations, which rely less on the retransmit timer. One result of TCP Reno's reduced use of the slow-start procedure [S94] is a decreased reliance on the retransmit timer. Newer TCP implementations with Reno-style congestion control algorithms are becoming widely available. Also, the use of intelligent congestion detection mechanisms in switches would reduce TCP's reliance on retransmit timers. Further, with congestion detection mechanisms such as those in RED algorithms, connections that use only a small fraction of the link bandwidth are unlikely to have packets dropped at the switch or gateway. In this case, packets are less likely to be dropped from connections with small windows, where the retransmit timer rather than the fast retransmit procedure might be required for recovery.

For TCP connections whose entire path is over ATM networks, reducing the duplicate ACK threshold from three to one can also improve performance by decreasing reliance on the retransmit timer. Because ATM networks should not reorder or duplicate cells, TCP connections transmitted over all-ATM networks do not necessarily have to wait for three duplicate ACKs to reliably infer a dropped packet, and could retransmit after one duplicate ACK.

C Packet Retransmission Rates

In the simulations of TCP over ATM with EPD, the effective throughput is generally high; the exceptions are simulations where the buffer contains only a few packets worth of cells. Nevertheless, the packet retransmission rate for the active TCP connections can be high even when the effective throughput on the congested link is close to optimal. The throughput can be high even while packet loss and retransmission rates are high because the throughput measures the efficiency of the link, how much "good" data is received across the link. Under high congestion, some connections experience dropped packets, while other packets from those connections are transmitted on the congested link. In this way the loss/retransmission rate can be high, while the overall link throughput is optimal.

Figure 7 shows the cell loss rates for simulations with TCP over plain ATM. Figures 8, 9, and 10 show the packet retransmission rates for simulations with TCP over plain ATM, TCP over ATM with EPD, and Packet TCP. The switch buffer size is shown on the x-axis; the y-axis shows the number of retransmitted packets, as a percentage of the total number of packets, disregarding duplicates. If packets are retransmitted more than once, the percentage retransmission can be over 100. For simulations with EPD, the cell loss rate is comparable to the packet retransmission rate.

As Figures 8 and Figure 9 show, Early Packet Discard reduces the number of retransmitted packets. However, even with Packet TCP the retransmission rate can be high. For TCP connections with small roundtrip times and large packets, the TCP window increase algorithm is fairly aggressive, and this can result in a significant fraction of dropped packets.

The important performance parameters for TCP traffic are not the cell loss rates but the average end-to-end packet delay (for interactive TCP traffic such as telnet traffic) and the total time for a connection to complete a transfer (for bulk-data TCP traffic such as FTP file transfers). These performance parameters are not unaffected by cell loss rates, but they depend at least as much on the effective throughput, average queue sizes, retransmission procedures of the transport protocol, and the underlying fairness of the network than on cell loss rates.

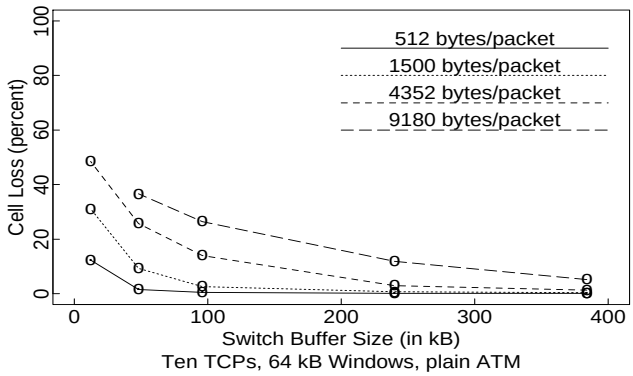


Figure 7: Cell Loss Rates for TCP over Plain ATM

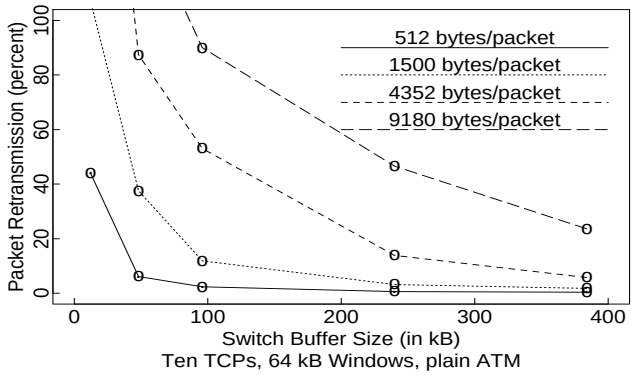


Figure 8: Packet Retransmission Rates for TCP over Plain ATM

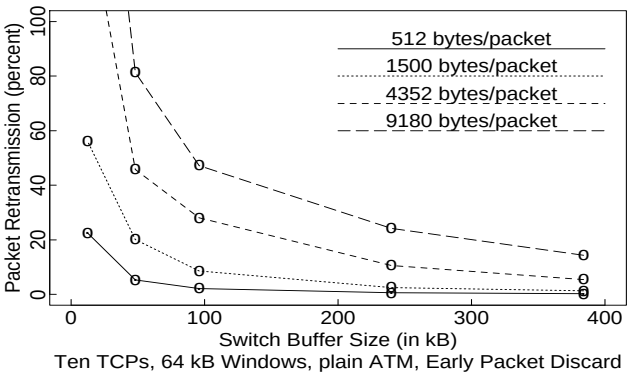


Figure 9: Packet Retransmission Rates for Early Packet Discard

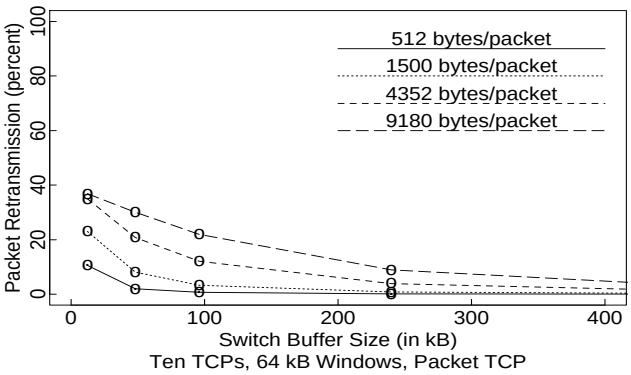


Figure 10: Packet Retransmission Rates for Packet TCP