Improved WWW Multimedia Transmission Performance in HTTP/TCP over ATM Networks

Ruay-Shiung Chang, *Member, IEEE,* and Chin-Dong Wang

***Abstract*— Transmission control protocol/Internet protocol (TCP/IP) is the de facto standard of the networking world. It dynamically adjusts routing of packets to accommodate failures in channels and allows construction of very large networks with little central management. But IP packets are based on the datagram model and are not really suited to real-time traffic. In order to overcome the drawbacks, a new network technology, ATM, is proposed. ATM provides quality of service (QOS) guarantees for various classes of applications and in-order delivery of packets via connection oriented virtual circuits. Unfortunately, when ATM is to be internetworked with the existing network infrastructure, some special signaling, addressing and routing protocols are needed. IP over ATM is one of the methods proposed by IETF. It allows existing TCP/IP applications to run on ATM end-stations and ATM networks to interconnect with legacy LAN/WAN technologies. But the performance of TCP/IP over ATM leaves something to be desired. Partial packet discard (PPD) and early packet discard (EPD) are two schemes to improve its performance.**

This paper proposes a "selective packet retransmission" scheme for improving HTTP/TCP performance when transmitting through ATM networks. In selective packet retransmission, we take advantage of the property of humans' perception tolerance for errors to determine whether to retransmit a corrupted TCP segment or not. For lossable data, such as images, when an error occurs because of cell losses, it will not be retransmitted. The simulations show that, for the same buffer size and traffic load, selective packet retransmission performs better than PPD, EPD, and plain TCP over ATM.

***Index Terms*—Early packet discard, hypertext transfer protocol, partial packet discard, quality of service, transmission control protocol.**

I. Introduction

INTERNET protocol (IP) is the most important network protocol because it is the de facto standard for communications across the world wide Internet. IP is a network layer protocol that provides connectionless delivery of data between two entities. Connectionless delivery means that the communication method does not guarantee delivery of packets between the source and destination. In an internetwork using IP, each datagram sent between the same two hosts can follow completely different paths. This dynamic traversal of the packets is dependent on many factors and is controlled by routers. A router dynamically determines the best path for a datagram to follow based on the status and congestion level of the network links.

On the contrary, the asynchronous transfer mode network is based on a connection-oriented model that uses virtual channels to enable fast synchronous or asynchronous data transfer. ATM requires that a virtual end-to-end connection be established before data transfer can begin. Virtual end-to-end connections are referred to as virtual channels. A virtual channel transports fixed length 53-byte cells between two endpoints. Each virtual channel consists of a VCI (virtual channel identifier) and a VPI (virtual path identifier) that is included in the 5-byte header of the ATM cell to provide identification for proper switching from an input port to an output port within an ATM switch. As the VC spans multiple links, the VCI can potentially change for each link. When a virtual channel is created, the ATM switch creates and maintains a table entry that maps inbound VCI's on an inbound port, to outbound VCI's on the outbound port. Due to its simplicity, the ATM switching can be easily implemented in hardware. Software is only required to manage the connections and maintain the switching table. ATM has many advantages over conventional IEEE 802 LAN's. For example, the flexibility to be used over a variety of physical mediums at a variety of speeds, the support of QoS, etc.

Unfortunately, when internetworking with the existing network infrastructure (predominantly TCP/IP), some special signaling, addressing and routing protocols are needed. There are two methods for implementing IP on an ATM network. The first is to implement a MAC layer to provide an emulation of IEEE 802 networks over ATM. The second approach replaces the data-link layer of the protocol stack with an ATM-aware layer. However, in either way certain features of ATM technology lead to substantial degradation of TCP/IP connection throughput under congestion conditions. For example, as cell size is fixed, the last cell will carry a padding (wasted space) and the 5-byte headers also increase the overhead. The poor performance of TCP/IP over ATM has been observed in previous studies [2], [21] and a number of remedies has been proposed [3], [11], [18], [21], [25]. Most of these mechanisms to improve TCP/IP performance over ATM networks require substantial complication of the ATM switches.

In this paper, we propose a new method to improve the HTTP/TCP performance over ATM networks, called the selective packet retransmission scheme. The principle of this method is to categorize the transfer data into two types lossable and unlossable. For lossable packets, they do not time out and retransmit when there are some errors. We use

Manuscript received February 4, 1999; revised June 25, 1999. The associate editor coordinating the review of this paper and approving it for publication was Dr. Anna Hac. This work was supported in part by the NSC under Contracts NSC87-2219-E-011-001 and NSC88-2213-E-259-007.

The authors are with the Department of Computer Science and Information Engineering, National Dong Hwa University, Hualien, Taiwan, R.O.C. (e-mail: rschang@csie.ndhu.edu.tw).

Publisher Item Identifier S 1520-9210(99)07423-4.

TABLE I
PACKET COUNT PERCENTAGES FOR VARIOUS PROTOCOLS ON NSFNET BACKBONE
(Source: ftp://ftp.merit.edu/statistics)

Month	HTTP	NNTP	FTP	Telnet	SMTP	DNS	#Packets(10^9)
1994 Jan.	1.5%	8.8%	21.4%	15.4%	7.4%	5.8%	55
1994 Apr.	2.8	9.0	20.0	13.2	8.4	5.0	71
1994 Jul.	4.5	10.6	19.8	13.9	7.5	5.3	74
1994 Oct.	7.0	9.8	19.7	12.6	8.1	5.3	100
1995 Jan	13.1	10.0	18.8	10.4	7.4	5.4	87
1995 Apr.	21.4	8.1	14.0	7.5	6.4	5.4	59

TABLE II
TCP PACKET COUNT PERCENTAGES ON vBNS
(source: http://www.vbns.org/nettraff/default.htm)

Quarter	HTTP+Web Cache	NNTP	FTP	Others
Jan. 1997 ~ Mar. 1997	36%	7%	10%	47%
Apr. 1997~Jun. 1997	58%	4%	20%	18%
Jul. 1997~Aug. 1997	11.7%	1.1%	6.1%	81.1%
Sep. 1997~Dec. 1997	Data not available			
Jan. 1998~Mar. 1998	29%	16%	42%	13%

the data transferred by HTTP (hypertext transfer protocol) to demonstrate the scheme. Intuitively, since HTTP data are now a major source of network traffic (see Table I for NSFNET data and Table II for vBNS Data),[1] when its performance improves, so will the performance of the whole network. Simulation results indeed corroborate with our expectations that performance can be improved with a little sacrifice in HTTP multimedia data (for example, graphic or image data) precision.

The remainder of this paper is organized as follows. Section II describes the past study aimed at improvement of TCP performance in high-speed ATM environment. The detail of our proposed mechanism is presented in Section III. The simulation assumptions, environments, and an analytic model are discussed in Section IV. Simulation results are illustrated in Sections V and VI concludes this paper.

II. RELATED WORK

Our proposed mechanism is a method for improving the HTTP/TCP performance over ATM networks. In the past, there have been many similar studies and they will be briefly described in the following.

In [18], the improvement of TCP performance is achieved through the use of large buffers in the ATM switch and through two simple methods: Usage parameter control (UPC) based and cell loss priority (CLP) based congestion control. The UPC-based congestion control algorithm uses a leaky bucket controller. Each forwarded cell consumes one token. Tokens that have not been consumed are stored in the finite-size token pool. A newly generated token is lost if the token pool is full. An arriving cell is tagged (its CLP bit is set to 1) if there is no token available at the time of its arrival. In effect, the leaky bucket limits the long-term average rate of the forwarded traffic to the token generating rate and limits the maximum burst size to the token pool size. The other method, CLP-based congestion control, uses a single CLP tagged packet as a probe for impending congestion when senders increase their offered load by raising their congestion windows. The discard of the CLP tagged packet, caused by buffer queues in excess of the CLP threshold, indicates possible congestion ahead. The sender than readjusts its TCP congestion window to maintain an average transmission rate close to the capacity of the bottleneck link.

The buffer management strategies described in [3], [21], and [25] have also been used to improve TCP performance. Partial packet discard [3] and early packet discard [21] schemes both reduce useless cells which congest the link. 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 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. It is shown that partial packet discard improves performance to a certain degree, but the throughput performance is still not optimal. To improve the performance further, a mechanism called early packet discard that brings throughput performance to its optimal level is proposed. In early packet discard, when the switch buffer queue reaches a threshold level, entire higher level data units (e.g., TCP/IP packets) are dropped.

Another method for discarding cells is drop from front [25]. When a cell arrives at a full buffer, the cell closest to being transmitted is dropped, thus creating space for the arriving cell. This policy causes duplicate acknowledgments of the previously received packets to be sent one whole buffer drain time earlier than is the case under tail drop. These quicker duplicate acknowledgments cause TCP with fast retransmit to recognize losses faster and invoke congestion control actions earlier than would be the case under tail drop. This earlier reaction translates into considerable performance improvement. Hence, drop from front successfully utilizes the ability of TCP with fast retransmit to quickly recognize and react to congestion information. Roughly, the earlier action by the sources causes the congestion not to grow quite as severe, which prevents later over-reaction by the sources, and thus increases throughput.

Performance of TCP or HTTP traffic over ATM has been studied in [8], [11], and [13]. Iliadias [11] investigates the performance of two schemes, threshold feedback and optimal feedback. They are applied hop-by-hop at the ATM layer and their objective is to ensure lossless operation. Cell losses are avoided by using simple “stop” and “start” signals sent from

[1] The NSFNET backbone was established in 1985 with great success. In 1995, NSF decommissioned the NSFNET backbone in April and let commercial Internet providers take over the role. At about the same time, a very high speed backbone network service (vBNS) was established by NSF for advanced research.

the receiving node to the transmitting node. The receiving node sends a "stop" signal to the transmitting node when the buffer content reaches a high-threshold H due to a cell arrival, and sends a "start" signal when the buffer contents have subsequently dropped below a low-threshold L due to a cell departure. Kalampoukas and Varma [13] study the TCP source policy and its effects on rate-controlled ATM networks. It is shown that the TCP slow-start phase can be significantly prolonged when the round-trip delay is small. Feldmann, Rexford, and Cáceres [8] use a continuous one-week trace of Internet traffic to evaluate the processor and switch overheads for transferring HTTP server traffic through a flow-switched network. The results show that moderate levels of HTTP traffic aggregation yield significant reductions in overhead with a negligible reduction in performance.

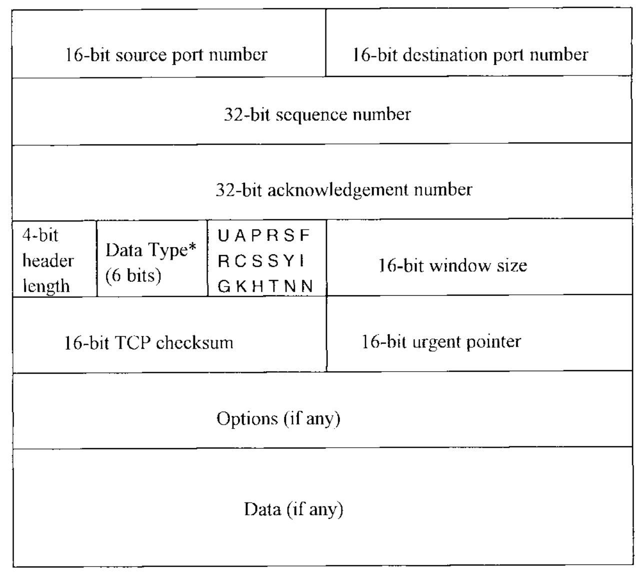
To improve the performance of IP over ATM switch, layer-3 switching is a possible method. However, a complete layer-3 switching is very complex and hard to implement. A compromise is to combine layer-2 switching for performance with IP network control for functionality and flexibility. There are a number of proposals in this direction, which include IP switching [17], the cell switch router [15], tag switching [20], ARIS [7], and MPLS [22]. A comparison of the above proposals can be found in [1].

III. Protocol Operations

In this section we will describe the modifications needed for HTTP and TCP protocols to reduce unnecessary retransmissions and improve packet latency. HTTP is a simple protocol. The client establishes a TCP connection to the server, issues a request, and reads back the server's response. The server denotes the end of its response by closing the connection. The client parses the HTML document according to tags and fetches the inlined files if necessary. According to the statistics in [4], GIF and JPEG files are the major source of HTTP traffic. These files are segmented into TCP packets for transmission. When the underlying networks are ATM, TCP packets are further sliced into ATM cells. If a cell is lost, due to buffer overflow or other reasons, in the ATM network, the whole packet containing this cell has to be retransmitted. This is a great waste of bandwidth and the slow start congestion control [26] algorithm of TCP makes it even worse. Therefore, it is appropriate to reconsider the wisdom of discarding the entire packet indiscriminately when only a few cells of the packet are lost. Especially in the World Wide Web application where many packets are part of a graphic, image, or audio/video file, the loss of a few cells only makes the data imperfect, but still discernible. This observation leads us to the idea of selective packet retransmission (SPR). When a packet is corrupted due to cell losses, this packet is not retransmitted if it belongs to files that can tolerate errors. To accomplish this purpose, we distinguish between two types of files. One can tolerate cell losses and can still be accepted by users. The other can tolerate no error, e.g., data files. Table III shows two values of classification. To access the inlined file with SPR in mind, the TCP connection between server and client should use the modified TCP header, shown in Fig. 1, to denote the

TABLE III
Data Type Classification

	URL File Formats (by Extension)	Data type value
Lossable data type	gif, jpg, bmp, png, tif, wav, mpg	000001
Unlossable data type	Others	000010



* This field is reserved in the original TCP header format.

Fig. 1. TCP header.

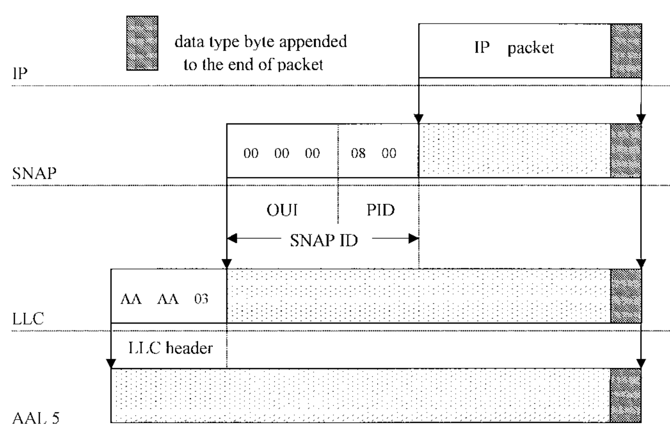


Fig. 2. IP packet encapsulation.

classification of the file transmitted. In case the cell containing TCP header is lost, the server should also append this data-type field to the trailer of the packet, shown in Fig. 2. Before AAL 5, each IP packet is prepended with 8-byte LLC/SNAP header. It includes 3-byte logical link control (LLC) field, 3-byte SNAP organizationally unique identifier (OUI), and 2-byte SNAP protocol identifier (PID) field. This PDU is translated to AAL5 cells and the data-type classification will fall into the last cell. Cells belonging to the same packet will pass through the ATM networks in order. When the destination receives the

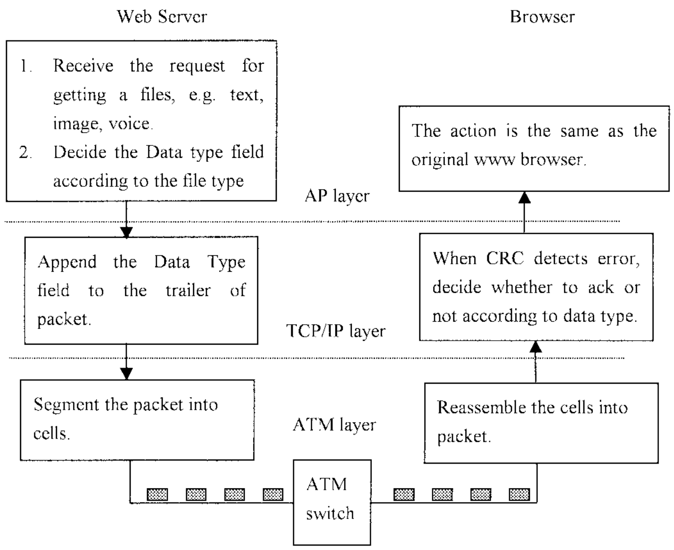


Fig. 3. The procedure of using selective packet retransmission.

Fig. 4. The original uncorrupted file in gif format: xmas.gif (8736 bytes).

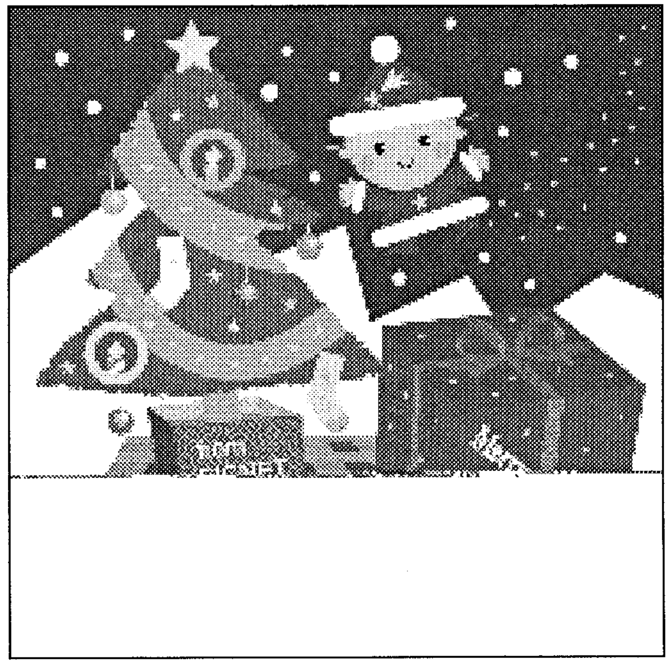


Fig. 5. The xmas.gif with 48 consecutive bytes deleted.

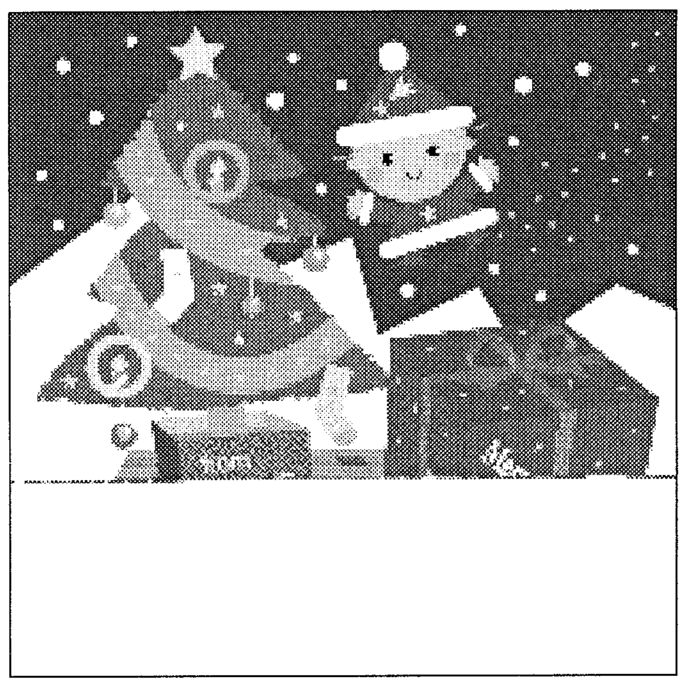


Fig. 6. The xmas.gif with $5 * 48$ consecutive bytes deleted.

last cell, it checks the data-type classification. When the data type classification is 000 001 and there is TCP checksum error, the receiving TCP would not discard this packet to prevent a retransmission and source congestion window adjustment. Please see Fig. 3 for the flowchart of protocol operations.

In the following, we use two image files (the first in gif format, the second in jpg) as examples. The correct images are shown in Figs. 4 and 8, respectively. Three cases are tested: the loss of 48 bytes (the size of a cell), of $5 * 48$ bytes, and of $20 * 48$ bytes. The results are shown in Figs. 5–7 and 9–11. It can be seen that the images, even though not perfect, are still acceptable to human eyes. An interesting research topic is to define a metric of acceptability for various media. In this paper, we use the definition that the corrupted file can still be recognized (displayed or played) by the corresponding media software. One exception is that if a lost cell is in the image header in gif format, the file will not be recognized by the LZW compression algorithm.

To further investigate the effects of cell losses in image files, many cases of cells lost are tested in Appendix A. The results show that in some cases even with a loss of 20 consecutive cells, the remaining data can still constitute a recognizable

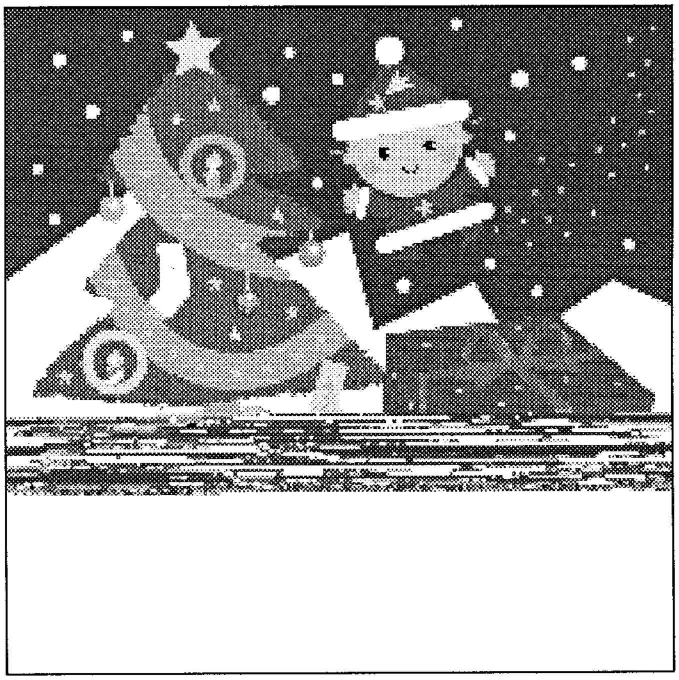


Fig. 7. The xmas.gif with $20 * 48$ consecutive bytes deleted.

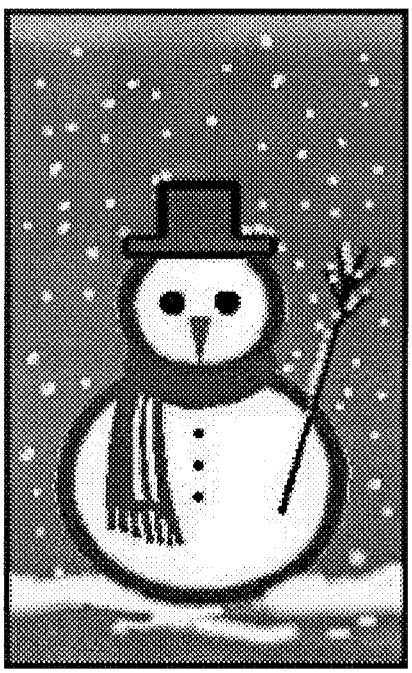


Fig. 8. Original uncorrupted jpg file: snow.jpg (18634 bytes).

picture. It is interesting to observe that jpg format is much more error-resilient than gif format.

In order to employ the selective packet retransmission scheme in the existing protocols (TCP/IP, HTTP), there are two things to be modified. First, the HTTP server must be able to add the classification information into the trailer of the last cell of the packet and use the modified TCP header. Second, the retransmission algorithm in the TCP layer must be modified to tolerate incomplete packets for some situations.

Fig. 9. The snow.jpg with 48 consective bytes deleted.

Fig. 10. The snow.jpg with $5 * 48$ consective bytes deleted.

IV. Simulations

A. Simulation Setup

The simulations use the network topology shown in Fig. 12. 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 round-trip 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 μs for each link.

Fig. 11. The snow.jpg with $20 * 48$ consective bytes deleted.

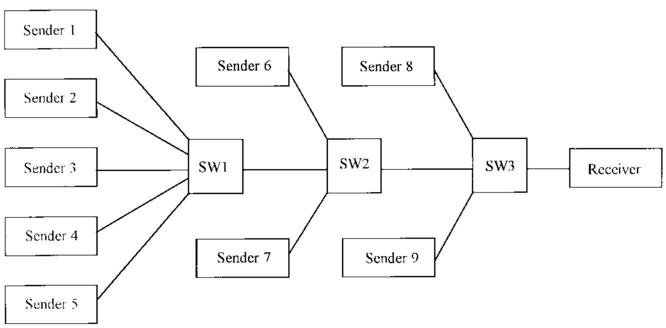


Fig. 12. Simulation scenario: the propagation delay is 3 μs and bandwidth is 141 Mbps for each link.

Simulations were run with nine "bulk-data" TCP connections. The number 9 is chosen to represent a relatively large number of simultaneous TCP's contending for the same resources. The simulation time is 10 s, which is reasonably long, relative to the average round-trip time, and is replicated 100 times. In the ATM switches the output ports use FIFO queuing. In the simulation, buffer size per output port is between 200 and 2000 cells.

To ensure a realistic representation of TCP/IP based traffic in the simulation, the TCP model includes the functionality of Jacobson's slow-start congestion control and avoidance scheme [26], but without Karn and Partridge's algorithm for improving round-trip estimates [14]. Since our simulation environment is very simple and round-trip time can be easily estimated in the long-term, our TCP model does not employ Karn and Partridge's algorithm. Jacobson's slow start congestion control algorithm is described as follows. In brief, slow-start requires each TCP connection to continuously probe the network trying to learn the amount of bandwidth currently available to the connection. The probing algorithm requires the sender to

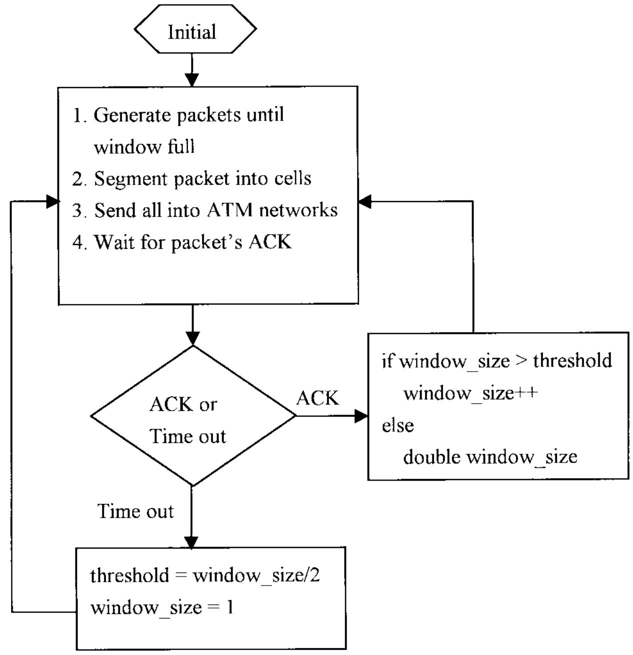


Fig. 13. Traffic generation flow chart.

keep a congestion window, which is estimated by a two-part algorithm. In the first part, the sender starts by sending one segment and waiting for the segment's acknowledgment. When the acknowledgment comes back, the sender sends two segments, and waits for their acknowledgment, after which the sender transmits four segments. This exponential growth of the congestion windows continues until the sender finds that a segment has been lost. When a segment is lost, the sender keeps a record of the size of the congestion window and starts the exponential sending pattern again from a window size of 1. But in this second stage, the sender grows the window exponentially only up to one-half the previous congestion window. After that, the sender grows the windows linearly.

Each sender uses an infinite source model at the application layer running on top of TCP and we assume 10%[2] of the data are lossable. This implies that TCP always has a packet to send as long as its windows permit it. The window size is determined by the status of ATM networks. Fig. 13 presents the traffic generator in this simulation. Other parameters used are

$$\text{TCP maximum segment size:} \quad \text{MSS} = 512, 1500 \text{ or } 4352 \text{ bytes}$$

$$\begin{aligned}\text{Time out period} \\ &= \text{Buffer size} * (\text{The number of switches in the path} + 1) \\ &\quad + \text{serialization delay.}\end{aligned}$$

In the past, the performance of ATM networks has been measured in cell-level metrics. Cell loss ratio (CLR), cell delay variation (CDV), and cell transfer delay (CTD) are some examples of cell-level metrics. Unfortunately, cell-level metrics

[2] In view of the multimedia rich web pages, 10% may be an underestimation. Please refer to http://www.gewis.win.tue.nl/conneg-bin/stats for a measurement of web data type. If the percentage were higher, the performance would be better.

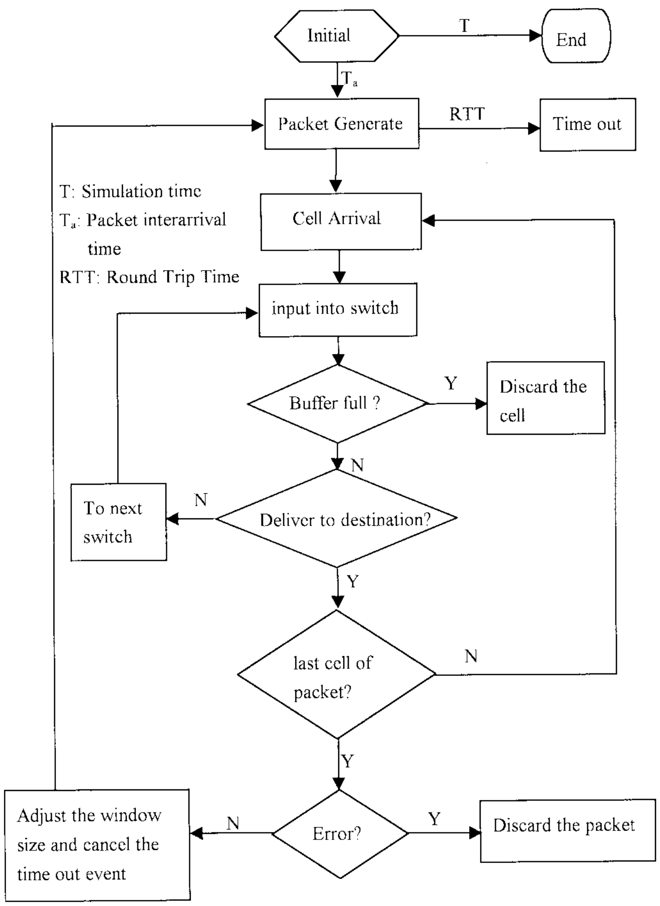


Fig. 14. Simulation flow for plain TCP over ATM.

do not reflect the performance required by the end users. For example, two users sending the same number of packets with the same cell loss rate might get different throughput due to the distribution of dropped cells among the packets. Hence, the user is interested in not only the cell loss rate but also the distribution of dropped cells among the packets. Therefore we define three metrics: effective throughput, packet loss rate, and packet delay. Effective throughput does not include cells that are part of a retransmission or an incomplete packet. Packet loss rate is defined as the fraction of packets that are not accepted by the application due to cell losses. Packet delay is defined as the packet round trip time.

B. Simulation Implementation

In this simulation, we want to compare the performances of four strategies: plain TCP over ATM, partial packet discard, early packet discard, and our selective packet discard. Fig. 14 is the flowchart of plain TCP over ATM. In PPD, the "buffer full?" condition is replaced with "cells belong to a corrupted packet or buffer full?" In EPD, the condition is replaced with "cells belong to a corrupted packet or buffer load reaches 80% when the first cell of a packet arrives at switch?" In SPR, the testing is the same with plain TCP over ATM. However, SPR will discard erroneous packets only if they are not multimedia data.

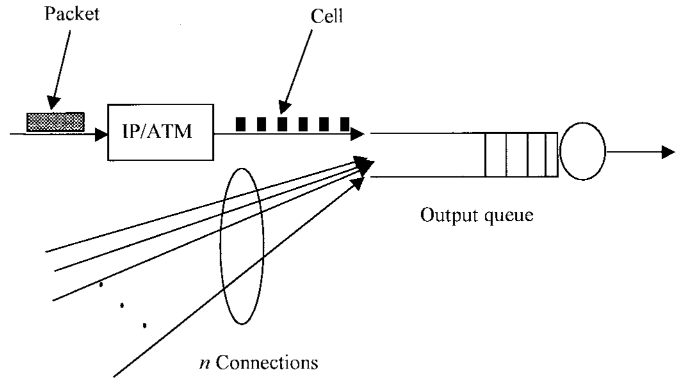


Fig. 15. Stochastic model.

In Fig. 14, "Initial" is the initialization state which does three tasks:

1) initialize parameters;
2) set the initial values for each state variable;
3) place the initial event notices into the future event list.

The initial event notices include the end event after time T and packet generate after T_a. When senders generate a packet, they also schedule a time out event after time RTT. The packet is then segmented into cells. When cells arrive at ATM switch, if the buffer is full, it will be discarded. When a cell is delivered to the destination and it is the last cell of a packet, the receiver will check if the packet is correct. If there are no errors, it will adjust the window size and cancel the time out event.

Before observing the simulation results, we may wonder what parameters will affect the packet loss rate. The following analysis shows that when the maximum TCP segment size increases, so will the TCP packet loss rate.

C. An Analysis

To transfer TCP/IP packets over ATM networks, packets have to be segmented into cells. To analyze the factors that affect the performance of this process, we consider the situation on a single output queue of an ATM switch. Assume that the cell length is fixed and time is slotted with slot size equal to a cell transmission time. Cells from a number of sources and destined to the same output link arrive simultaneously at the beginning of a time slot and are stored in the buffer, shown in Fig. 15. The first cell in the queue is then immediately served at the end of that time slot.

We define IP/ATM traffic pattern in the following. Assume

1) the period between packet is exponentially distributed;
2) average packet length is l cells;
3) capacity of output buffer is c cells.

The transition probabilities for the IP/ATM connection are then

$$\begin{array}{ccc} & 0 \quad 1 & \leftarrow \text{ Cell number for next slot} \\ \Phi = \begin{array}{c} 0 \\ 1 \\ \uparrow \end{array} & \left(\begin{array}{cc} 1-\lambda/t & \lambda/t \\ 1/l & 1-1/l \\ & \end{array}\right) & \end{array}$$

Cell number for previous slot.

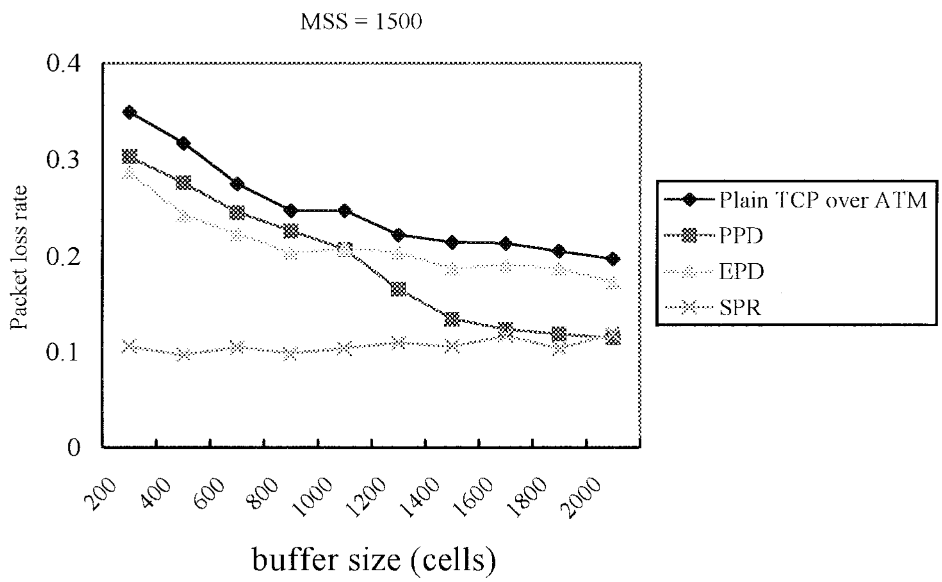


Fig. 16. Packet loss rate for MSS = 512 bytes.

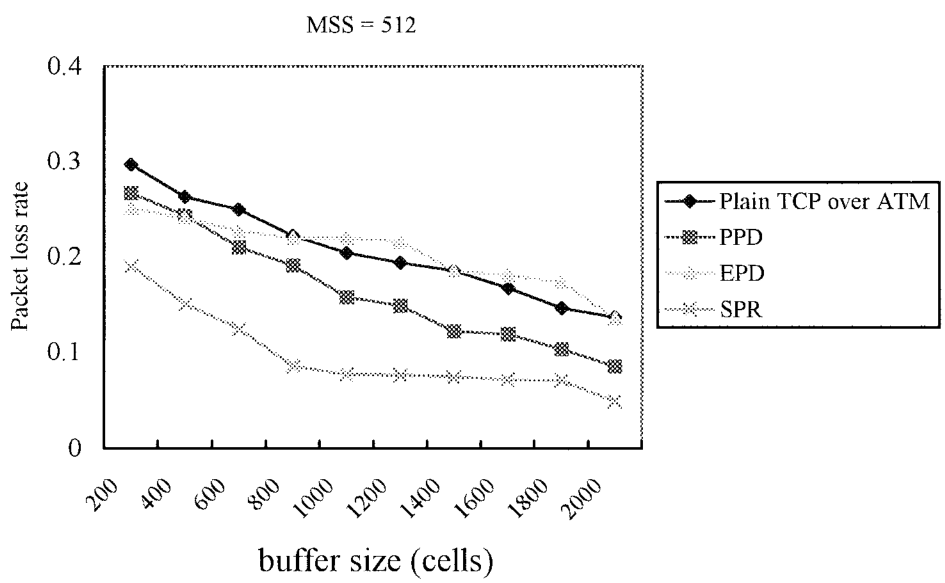


Fig. 17. Packet loss rate for MSS = 1500 bytes.

where λ is the number of cells arrived during time period t. If we assume the burst length of cells is equal to the average packet length l and is geometrically distributed, we have the above probabilities in the second row.

When the IP/ATM connection is at the equilibrium state, we have

$$\begin{cases} [P_0 \quad P_1] = [P_0 \quad P_1] \cdot \begin{bmatrix} 1-\dfrac{\lambda}{t} & \dfrac{\lambda}{t} \\ \dfrac{1}{l} & 1-\dfrac{1}{l} \end{bmatrix} \\ P_0 + P_1 = 1 \end{cases}$$

where P_1 is the probability of the event that one cell arrived in a time slot for the IP/ATM connection and P_0 is the probability of the event that no cells arrived. Solving the equations, we have

$$P_1 = \frac{l \cdot \lambda}{l \cdot \lambda + t}$$
$$P_0 = 1 - P_1 = \frac{t}{l \cdot \lambda + t}.$$

For the other n connections, let Y be the random variable representing the total number of cells arrived in a time slot and assume it has a Poisson distribution with mean μ.

If Z is the total number of cells arrived, including IP/ATM connection and the other n connections, at a time slot, then

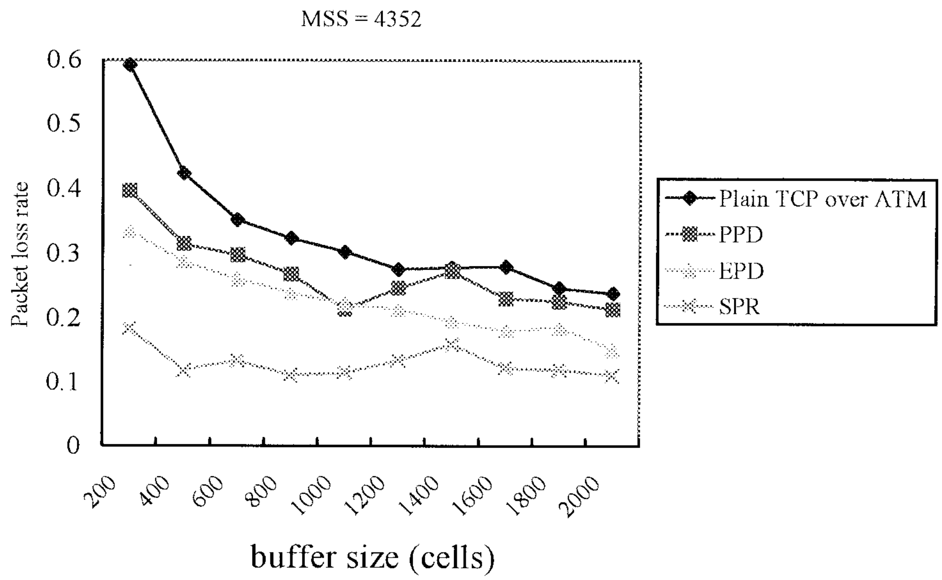


Fig. 18. Packet loss rate for MSS = 4352 bytes.

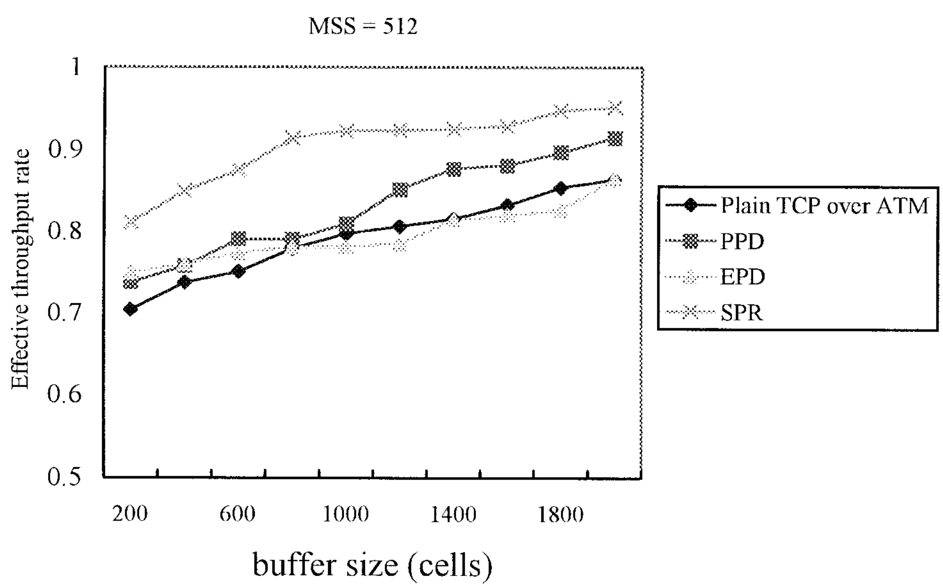


Fig. 19. Effective throughput rate for MSS = 512 bytes.

either $Z = Y + 1$ (IP/ATM connection has one cell) or $Z = Y$. The range of Z is between 0 and $n + 1$ cells. With the above, the probability function of Z is

$$\begin{aligned} P(Z = x) &= P(\text{IP/ATM connection has 1 cell}) \\ &\quad + P(\text{IP/ATM connection has no cell}) \\ &= P_1 * P(Y = x - 1) + P_0 * P(Y = x) \\ &= \frac{l \cdot \lambda}{l \cdot \lambda + t} * \frac{\mu^{x-1} e^{-\mu}}{(x-1)!} + \frac{t}{l \cdot \lambda + t} * \frac{\mu^x e^{-\mu}}{x!} \\ &= \frac{(l \cdot \lambda \cdot x + \mu \cdot t) \cdot \mu^{x-1} \cdot e^{-\mu}}{(l \cdot \lambda + t) x!}. \end{aligned}$$

Let random variable B stand for the number of vacancies in the buffers. Since there are c buffers, a cell will be lost if the number of cells arrived in a slot is greater than the number of vacancies available by at least 2 (since one cell will leave the switch at a time slot). Therefore, the probability of cells lost, $P(\text{lost})$ is

$$\begin{aligned} P(\text{lost}) &= P(Z=2, B=0) + P(Z = 3, B < 2) \\ &\quad + P(Z = 4, B < 3) + \cdots + P(Z = n + 1, B < n). \end{aligned}$$

Assume the number of buffer vacancies and the events of cell arrival are independent of each other. Suppose the buffers have equal probabilities for being at the state of 0 vacancy, 1

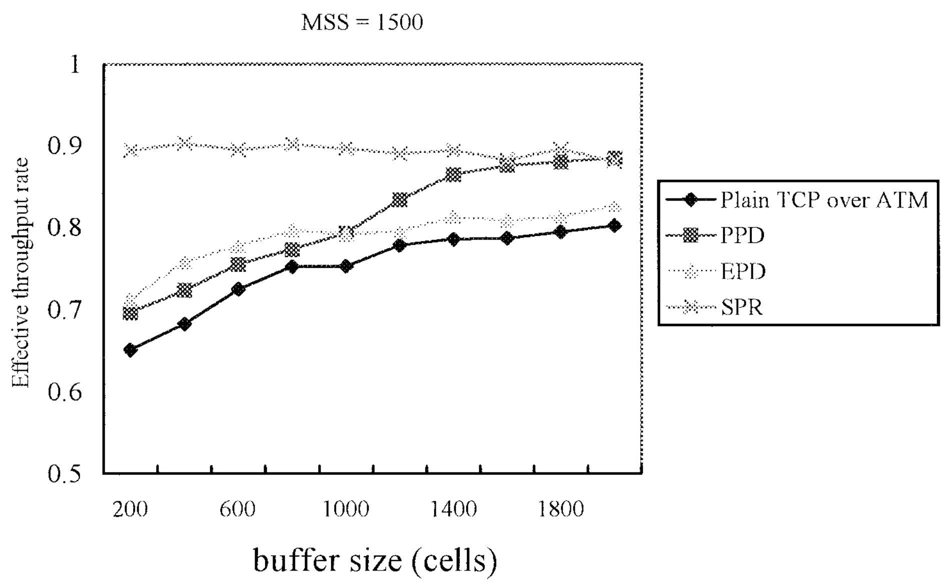


Fig. 20. Effective throughput rate for MSS = 1500 bytes.

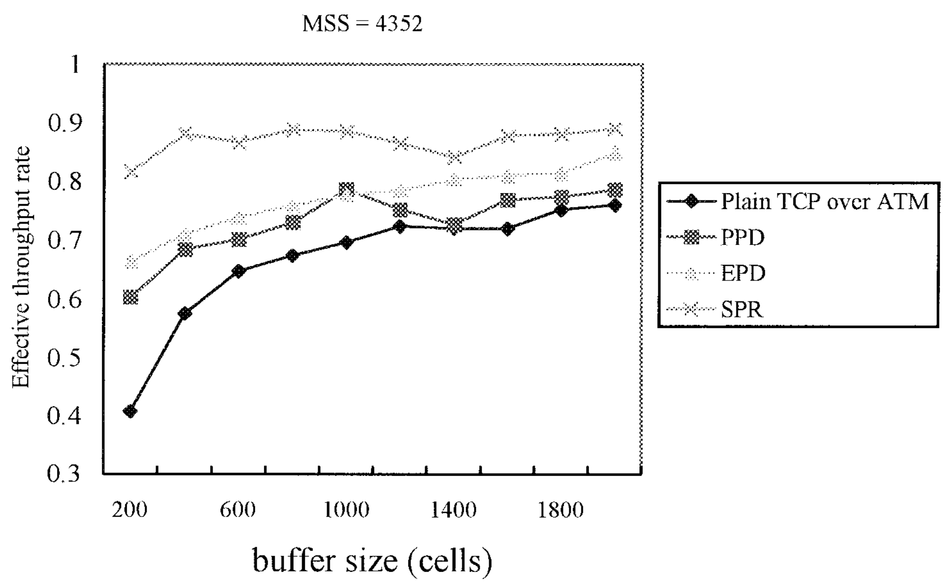


Fig. 21. Effective throughput rate for MSS = 4352 bytes.

vacancy, 2 vacancies, and so on. Then

$$\begin{aligned}P(\text{lost}) &= P(Z=2)P(B=0)+P(Z=3)P(B<2)\\ &\quad +\cdots+P(Z=n+1)P(B<n)\\ &= \sum_{i=2}^{n+1} P(Z=i)P(B<i-1)\\ &= \sum_{i=2}^{n+1} P(Z=i)[P(B=0)+P(B=1)\\ &\quad +\cdots+P(B=i-2)]\\ &= \sum_{i=2}^{n+1} P(Z=i)\cdot\frac{i-1}{c+1}\\ &= \frac{1}{c+1}\sum_{i=2}^{n+1}(i-1)\cdot\frac{(l\cdot\lambda\cdot i+\mu\cdot t)\cdot\mu^{i-1}\cdot e^{-\mu}}{(l\cdot\lambda+t)\cdot i!}\\ &= \frac{l\lambda\cdot e^{-u}}{(c+l)(l\lambda+t)}\sum_{i=2}^{n+1}\frac{\mu^{i-1}}{(i-2)!}+\frac{te^{-\mu}}{(c+1)(l\lambda+t)}\\ &\quad \cdot\sum_{i=2}^{n+1}\frac{(i-1)\mu^i}{i!}\\ &= \frac{l\lambda\cdot e^{-\mu}\mu}{(c+1)(l\lambda+t)}\sum_{i=0}^{n-1}\frac{\mu^i}{i!}+\frac{te^{-\mu}}{(c+1)(l\lambda+t)}\\ &\quad \cdot\left[\mu\left(\sum_{i=0}^{n}\frac{\mu^i}{i!}-1\right)-\left(\sum_{i=0}^{n+1}\frac{\mu^i}{i!}-\mu-1\right)\right].\end{aligned}$$

If n approaches infinity, we have

$$P(\text{lost})_{n\to\infty}=\frac{l\lambda\mu+te^{-\mu}+t\mu-t}{(c+1)(l\lambda+t)}.$$

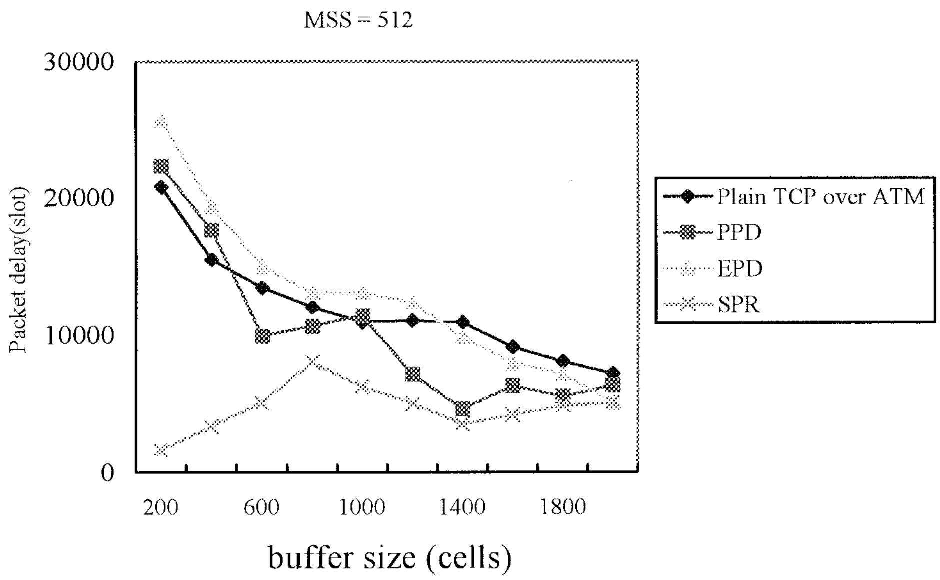


Fig. 22. Packet delay for MSS = 512 bytes.

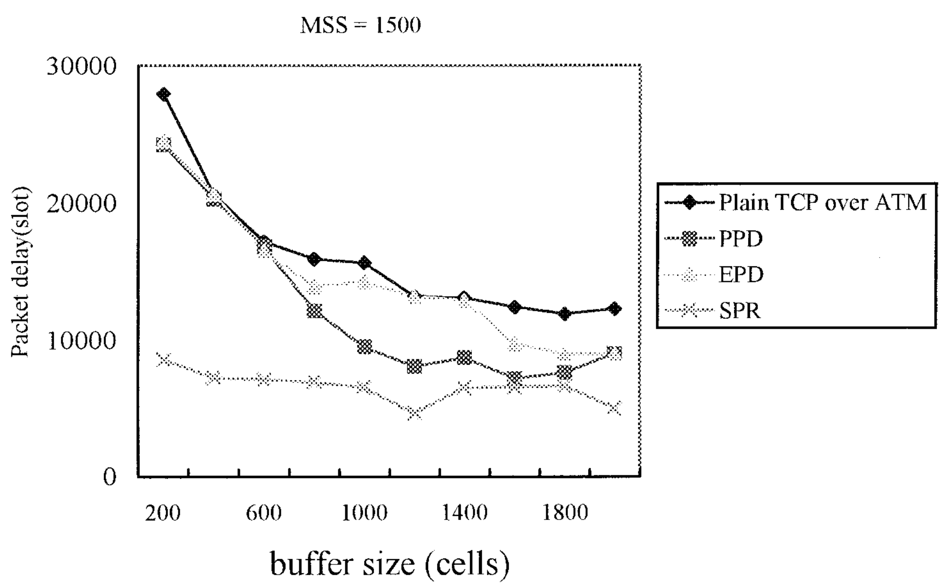


Fig. 23. Packet delay for MSS = 1500 bytes.

As n approaches infinity, we can assume that the arrived cells per slot are more than the buffers can handle ($\mu > c$). Therefore, when l (average packet length) increases, so will the packet lost probability. The simulations in the next section corroborate this when we increase the TCP segment size. Also it is obvious that when c (the number of buffers) increases, the packet lost probability will decrease.

V. Simulation Results

In this section we present the simulation results for the four TCP over ATM schemes, which are plain TCP over ATM, partial packet discard, early packet discard, and selective packet retransmission. The buffer sizes in the following plots refer to the available buffer size per output port for each of the switches. In early packet discard mechanism we assume that the threshold value is 80% of buffer size.

Fig. 16 is the packet loss rate for various buffer sizes. The MSS (TCP maximum segment size) is 512 bytes.

In this figure, note that SPR is better than PPD, EPD and plain TCP over ATM and packet loss rate decreases with increasing buffer size. The curve of EPD is the worst in some buffer sizes. This is because the threshold (80%) of buffer size may not be the optimal value for that buffer size.

Figs. 17 and 18 display the results for different MSS. When the MSS increases, the number of cells produced by segmenting the packet also increases. If one of the cells is lost, the whole packet is deemed to be lost. However, plain TCP over ATM does not react to this situation. Therefore, it has higher packet loss rate than other schemes.

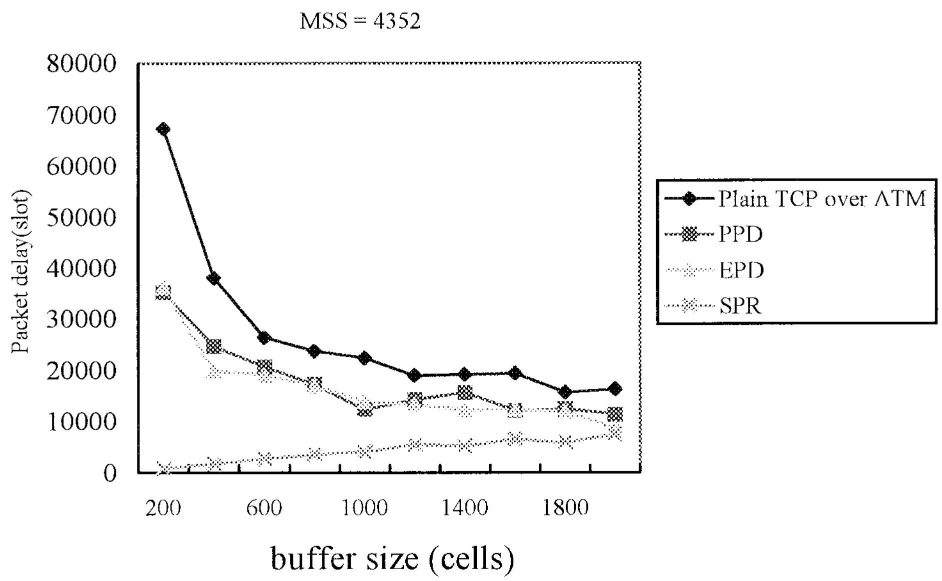


Fig. 24. Packet delay for MSS = 4352 bytes.

From Figs. 19–21, we show the effective throughput rate for various buffer sizes. Observe that schemes with a bigger buffer size have a higher effective throughput rate. SPR is the best scheme and PPD is better than plain TCP over ATM.

The average packet delay for connections 1–5 in Fig. 13 is shown in Figs. 22–24. If the buffer size is small, the cell loss rate will increase. Hence the packet loss rate will also increase. Lost packets will have to be retransmitted. This will in turn increase the packet delay time. However, in SPR schemes, some lost cells do not affect the acceptance of the corresponding packet. Therefore, it has the lowest average packet delay.

TABLE IV
THE EXPERIMENT OF JPG FORMAT FILE

Continuous # of cells lost in a file	Sample number	Visible number	Visible rate
1	251	231	0.92
5	251	231	0.92
10	251	231	0.92
20	251	231	0.92

TABLE V
THE EXPERIMENT OF GIF FORMAT FILE

Continuous # of cells lost in a file	Sample number	Visible number	Visible rate
1	121	60	0.495
5	121	55	0.454
10	121	51	0.421
20	121	45	0.371

VI. CONCLUSIONS

In this paper, we propose using "selective packet retransmission" policy to improve the performance of HTTP/TCP over ATM. The motivation was to exploit the property of humans' perception tolerance for imperfect multimedia data. When a packet is corrupted due to cell losses, this packet is not retransmitted in TCP if it belongs to graphic or image files. We conduct an experiment to test the various file types for their lossabilities. We also compared TCP performance under selective packet retransmission to that with PPD and EPD. PPD and EPD can improve the performance of TCP over ATM. But the implementation of PPD requires the switch to keep additional per VC state in order to recognize which VC's want to use PPD and the implementation of EPD requires the switch to monitor the active buffer queue size. Selective packet retransmission not only has better performance than PPD and EPD, but also doesn't add complexities to ATM switches.

People use the World Wide Web because it gives quick and easy access to a tremendous variety of information in remote locations. Users do not like to wait for their results. That is, users care about Web latency. However, the Web is quite rich in graphics. According to statistics in [4], over 50% of all pages contain as least one image reference. These images consume many resources of networks. For many of the pages that contain a large number of images, our method improves the latency of images and reduces the ATM switches processing. This will enhance the performance of other TCP applications in ATM networks.

In the future, we will explore more multimedia file types for error tolerance and model more realistic HTTP traffic and other TCP traffic for simulations. Another possible research direction is to design an Internet-aware ATM switch. The buffering strategy and the queuing policy should all be considered with the packets in mind. This is especially important if ATM is to become the savior of the congested Internet. Unless an efficient and fast global network infrastructure for Internet is designed and implemented, the world wide wait phenomenon may persist for years to come.

APPENDIX A

We conduct an experiment which deletes some continuous cells in a graphic or image file, and view the results of corrupted picture via Paint Shop Pro. The results are summarized in Tables IV and V. The visible number in both tables means the number of corrupted files that can still be displayed by the Paint Shop Pro program.

ACKNOWLEDGMENT

The authors would like to thank the editor and the anonymous referees for their comments that greatly improved the quality of this paper.

REFERENCES

[1] S. Agrawal, "IP Switching," [Online.] Available http://www.cis.ohio-state.edu/~jain/cis788 97/ip_switching/index.htm.
[2] G. Armitage and K. Adams, "How inefficient is IP over ATM anyway?," *IEEE Network*, Jan./Feb. 1995, pp. 15–26.
[3] ——, "Packet reassembly during cell loss," *IEEE Network*, vol. 7, no. 5, pp. 26–34, Sept. 1993.
[4] T. Bray, "Measuring the web," *Computer Networks and ISDN System*, vol. 28, no. 13, pp. 993–1005, Oct. 1996.
[5] J. Calvignac, J. Cherbonnier, I. Iliadis, J.-Y. Le Boudec, and D. Orsatti, "ATM best-effort service and its management in the LAN," in *Proc. EFOC&N'94*, Heidelberg, Germany, June 1994, pp. 155–160.
[6] P. B. Danzig, S. Jamin, R. Caceres, D. J. Mitzel, and D. Estrin, "An empirical workload model for driving wide-area TCP/IP network simulation," *J. Internetworking*, 1992.
[7] N. Feldman and A. Viswanathan, ARIS specification, Internet draft, draft-feldman-aris-spec-00.txt, Mar. 1997. [Online.] Available-http://www.networking.ibm. com/isr/arisspec.html.
[8] A. Feldmann, J. Rexford, and R. Caceres, "Efficient policies for carrying web traffic over flow-switched networks," *IEEE/ACM Trans. Networking*, vol. 6, Dec. 1998, pp. 673–685.
[9] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Trans. Networking*, vol. 1, pp. 397–413, Aug. 1993.
[10] D. Gaiti and G. Pujolle, "Performance management issues in ATM networks: Traffic and congestion control," *IEEE/ACM Trans. Networking*, vol. 2, pp. 249–257, Apr. 1996.
[11] I. Iliadis, "Performance of TCP traffic and ATM feedback congestion control mechanisms," in *IEEE Global Telecommunications Conf.*, vol. 3, pp. 1930–1934, Nov. 1995.
[12] R. Jain, "Congestion control and traffic management in ATM networks: Recent advances and a survey," *Computer Networks and ISDN System*, vol. 28, no. 13, pp. 1723–1738, Oct. 1996.
[13] L. Kalampoukas and A. Varma, "Analysis of source policy and its effects on TCP in rate-controlled ATM networks," *IEEE/ACM Trans. Networking*, vol. 6, pp. 599–610, Oct. 1998.
[14] P. Karn and C. Partridge, "Improving round-trip time estimates in reliable transport protocols," in *SIGCOMM'87, Computer Communications Review*, vol. 17, no. 5, Aug. 1987.
[15] Y. Katsube, K. Nagami, and H. Esaki, "Tohiba's router architecture extensions for ATM: Overview," in *IETF RFC 2098*, Feb. 1997.
[16] S. C. Liew, "Performance of various input-buffered and output-buffered ATM switch design principles under bursty traffic: Simulation study," *IEEE Trans. Communications*, vol. 42, nos. 2/3/4, pp. 1371–1379, 1994.
[17] P. Newman, G. Minshall, and T. L. Lyon, "IP switching-ATM under IP," *IEEE/ACM Trans. Networking*, vol. 6, no. 2, pp. 117–129, Apr. 1998.
[18] M. Perloff and K. Reiss, "TCP performance enhancements in ATM networks with large buffers and priority mechanisms," in *ICC'95 Seattle Communications—Gateway to Globalization*, vol. 3, pp. 1478–1482.
[19] V. N. Padmanabhan and J. C. Mogul, "Improving HTTP latency," *Computer Networks and ISDN System*, vol. 28, no. 13, pp. 25–35, Oct. 1996.
[20] Y. Rekhter, B. Davie, D. Katz, E. Rosen, and G. Swallow, "Cisco systems' tag switching architecture overview," in *IETF RFC 2105*, Feb. 1997.
[21] A. Romanow and S. Foyd, "Dynamics of TCP traffic over ATM network," *IEEE J. Select. Areas Commun.*, May 1995.
[22] E. C. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol label switching architecture," Internet draft, draft-ietf-mpls-arch-05.txt, Apr. 1999.
[23] W. R. Stevens, *TCP/IP Illustrated: The Protocols.* Addison-Wesley, vol. 1.
[24] W. R Stevens, *TCP/IP Illustrated: TCP for Transactions, HTTP, NNTP, and the UNIX Domain Protocols.* Reading, MA: Addison-Wesley, vol. 3.
[25] T. V. Lankshman, A. Neidhardt, and T. J. Ott, "The drop from front strategy in TCP and in TCP over ATM," in *Proc. IEEE INFOCOM'96*, vol. 3, pp. 1242–1250.
[26] V. Jacobson, "Congestion control and avoidance," in *SIGCOMM'88, Computer Communications Review*, Sept. 1988, vol. 18, no. 4, pp. 314–329.

Ruay-Shiung Chang (M'98) received the B.S.E.E. degree from the National Taiwan University, Taipei, Taiwan, R.O.C., in 1980 and the Ph.D. degree in computer science from National Tsing Hua University, Hsinchu, Taiwan, in 1988.

From 1988 to 1992, he was a Senior Researcher at the Chung Shan Institute of Science and Technology, Taiwan. From 1992 to 1998, he was with the Department of Information Management, National Taiwan University of Science and Technology. In August 1998, he joined the Department of Computer Science and Information Engineering, National Dong Hwa University, Hualien, Taiwan, where he is currently the Chairman. His research interests include Internet, high-speed networks, and wireless networks.

Dr. Chang is a member of ACM and the R.O.C. Institute of Information and Computing Machinery.

Chin-Dong Wang received the B.S. and M.S. degrees in information management from National Yunlin University of Science and Technology, Taiwan, R.O.C., and National Taiwan University of Science and Technology in 1995 and 1997, respectively. He currently works for a company in Taiwan. His main research interests include high-speed networks, Internet, and network management.