INTERNATIONAL JOURNAL OF COMMUNICATION SYSTEMS
*Int. J. Commun. Syst.* **12**, 309–319 (1999)

# The dual flow control problem of TCP over ATM ABR services

Yuan-Cheng Lai[1]*, Ying-Dar Lin[2] and Hsiu-Fen Hung[2]

[1] *Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan*
[2] *Department of Computer Information Science, National Chiao Tung University, Hsinchu, Taiwan*

## SUMMARY

In this paper, we investigate the dual control problem—TCP flow control at the TCP layer and ABR flow control at the ATM layer. First, we observe that TCP flow control and ABR flow control cannot co-operate well. The worst case is that the slow start after packet loss causes high but unused ACR (Allowed Cell Rate) which raises the potential of cell loss and an underflowed switch queue which reduces ABR throughput. We suggest to implement a use-it-or-lose-it policy for ABR and fast recovery for TCP to avoid these phenomena. Copyright © 1999 John Wiley & Sons, Ltd.

KEY WORDS: TCP; ATM; ABR; flow control

## 1. Introduction

Asynchronous transfer mode (ATM) is the most promising transfer technology for implementing B-ISDN (broadband integrated service digital network). However, today's Internet environment is based on TCP/IP. Hence, combining the virtues of both,[1] the TCP/ATM protocol stack is shown in Figure 1.[2]

The transfer unit of TCP is a variable-size packet(segment); the transfer unit of ATM is a fixed-size cell. TCP passes the packet to IP layer to be IP datagrams. ATM adaptation layer(AAL) segments IP datagrams into cells, passes them to ATM layer for transmission using the ABR (available bit rate) or UBR (unspecified bit rate) services.

ATM provides UBR and ABR service categories for data transfer. The ABR service is intended to fully utilize the available bandwidth. A flow control mechanism is specified to control the source rate in response to the changing condition of the ATM layer. The UBR service however does not have a flow control mechanism. When congestion occurs, discarding cells at the switches is the only response.

Many studies investigated the performance of TCP over ATM with UBR or ABR service. Several researchers have identified the poor performance of TCP over ATM with UBR ser vice.[2–7] This is largely due to the fact that the loss of a single ATM cell means the entire TCP segment is effectively lost, thus the bandwidth to transmit the remaining cells of this segment is

* Correspondence to: Yuan-Cheng Lai, Department of Computer Science and Information Engineering, National Cheng Kung University, No. 1 University Road, Tainan, Taiwan.

CCC 1074–5351/99/040309–11$17.50
Copyright © 1999 John Wiley & Sons, Ltd.
*Received April 1998*
*Accepted April 1999*

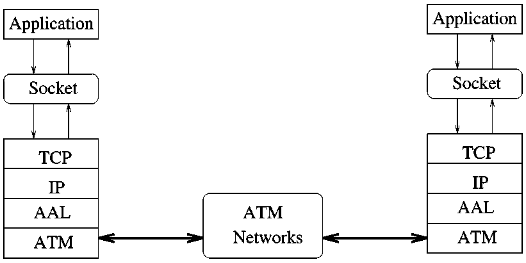


Figure 1. TCP/ATM protocol stack

wasted. Since UBR does not have a flow control mechanism, cell loss is inevitable. Allyn Romanow and Sally Floyd proposed the early packet discard and the partial packet discard schemes to prevent cells of the corrupted packets from being transmitted.[6]

In some cases TCP achieves better performance with ABR service than with UBR service plus the early packet discard scheme.[8–10] ABR service provides fair bandwidth allocation and high link utilization and requires a relatively small switch buffer in a LAN environment.[8] In a WAN environment with a large propagation delay, the performance degrades due to the cell loss caused by the delayed adjustment of source rate. Meanwhile, TCP will start its complex congestion control algorithm when it detects the packet loss. This attracts researchers to investigate the dual congestion control, i.e. TCP flow control over ABR flow control.[11–16] Some authors[13–15] proposed to enhance TCP congestion control mechanism using binary congestion notification (BCN). With this scheme, switches inform the sources about their congestion state by setting a congestion bit in the data packets. Other studies kept both TCP flow control and ABR flow control intact. They investigated the effect of various factors on TCP throughput and fairness.[9,11] The factors that have been examined are TCP timer granularity, switch buffering, ABR parameters and the cell drop policy at the switches.

In this paper, we investigate the time-dependent behaviour of these two flow-control mechanisms and evaluate their interaction. We identify and describe the asynchronous phenomena which causes buffer overflow and underflow. Some suggestions will be given to improve the performance. We also study the effect of various parameters on the performance. The parameters examined are maximum segment size, receiver buffer size, and rate increase factor. We use the finalized ABR flow control version that was published in April 1996.[16] Many researches were based on the old version.

Section 2 describes the TCP flow control and ABR flow control briefly. The simulation model and parameters are given in Section 3. Section 4 depicts the effects and suggestion of TCP over ABR. Section 5 gives the conclusion and future work.

## 2. Overview

### 2.1. *TCP flow control*

TCP flow control is based on the sliding window with a variable window size.[17] Each time an acknowledgement is received, the TCP end system sets the TCP window as the minimum of the advertisement window and the congestion window(*cwnd*). The advertisement window specifies

Copyright © 1999 John Wiley & Sons, Ltd.

the additional octets that the receiver can receive without overflowing the receiver buffer. The sender performs slow start and congestion avoidance algorithm to maintain *cwnd*. When starting a connection *cwnd* is initialized to one packet. *Cwnd* is then increased by one packet, each time when an acknowledgement is received. This is the slow start algorithm. After the TCP window is larger than *ssthresh* (a slow start threshold), the congestion avoidance process is performed, where *cwnd* is only increased by 1/*cwnd* packet each time. *Ssthresh* is initialized to 65536 bytes which is the maximum window size of TCP. When a packet is lost, one-half of the current TCP window is saved in *ssthresh*, and the slow start process is done again.

Each time when the sender sends a packet, it starts a retransmission timer. It is important to set the retransmission timeout value which is used to detect the packet loss. If the value is set too long, the performance degrades due to delayed awareness of the packet loss. If it is set too short, the sender will perform unnecessary retransmissions. TCP estimates the retransmission timeout based on the measured round trip time. The details can be found in Reference 17.

In addition to the expiration of the retransmission timer, the duplicate acknowledgements can be used to detect the loss of a packet. When three or more duplicate acknowledgements are received by the sender, it is a strong indication that a packet has been lost. The sender performs a retransmission of what appears to be the missing packet, without waiting for the retransmission timer to expire. This is the fast retransmission scheme.[18] Next, the congestion avoidance, instead of slow start, is performed. This is the fast recovery.[18]

There are three parameters which influence the network performance, namely:

- Maximum segment size (MSS): MSS refers to the amount of data that a source can transmit at one time.
- Receiver buffer size (Wrcv): Basically, the receiver buffer size must be at least as large as the product of available bandwidth to this connection and delay to achieve maximum utili zation.
- Clock granularity (Grain): The current TCP algorithm uses a clock granularity of 300–500 ms to measure the round-trip-time. It is too coarse in a high-speed low-propagation delay ATM environment. Allyn Romanow suggested to set it to 0·1 ms,[6] but Kalyaanaraman suggested 100 ms.[11]

### *2.2. ABR flow control*

We now briefly introduce the basic operation of the rate-based control mechanism.[16] When a virtual channel (VC) is established, the source end system (SES) sends cells at the allowed cell rate (ACR) which is set at the initial cell rate (ICR). In order to probe the congestion status of the network, the SES sends a forward resource management (RM) cell every *Nrm* data cells. Each switch may set certain fields of the RM cell to indicate its own congestion status or the bandwidth the VC source should use. The destination end system (DES) returns the forward RM cell as a backward RM cell to the SES. According to the received backward RM cell, the SES adjusts its allowed cell rate, which is bounded between peak cell rate (PCR) and minimum cell rate (MCR).

The RM cell contains a 1-bit congestion indication (CI) which is set to zero, and an explicit rate (ER) field which is set to PCR initially by the SES. When the SES receives a backward RM cell, it modifies its ACR using additive increase and multiplicative decrease. Depending on CI and ER

Copyright © 1999 John Wiley & Sons, Ltd.

fields in RM cells, the new ACR is computed as

$$\mathrm{ACR} = \max(\min(\mathrm{ACR} + \mathrm{RIF} * \mathrm{PCR}, \mathrm{ER}), \mathrm{MCR}) \quad \text{if CI} = 0$$

$$\mathrm{ACR} = \max(\min(\mathrm{ACR} * (1 - \mathrm{RDF}), \mathrm{ER}), \mathrm{MCR}) \quad \text{if CI} = 1$$

where RIF is the rate increase factor and RDF is the rate decrease factor.

According to the way of congestion monitoring and feedback mechanism, various switch mechanisms are proposed.[16] In our simulation, we use an EPRCA (enhanced proportional control algorithm) switch mechanism.[16]

EPRCA is an explicit rate marking switch mechanism. It supports intelligent marking, during congestion, to selectively mark certain VCs for a rate reduction, rather than all VCs. The switch has two thresholds of queue length: the congested threshold ($Q_{\mathrm{L}}$) and the very congested threshold(DQT) to determine the state of the network. The switch computes a mean allowed cell rate(MACR) for all VCs. The MACR is initialized to initial rate for MACR(IMR). When the switch receives the RM cell from the source, it computes MACR by $\mathrm{MACR} = \mathrm{MACR} + (\mathrm{ACR} - \mathrm{MACR}) * \mathrm{AV}$ when either it is in the congested state and ACR $<$ MACR or it is not in the congested state and ACR $>$ MACR$*$VCS, where AV is the exponential averaging factor and VCS is the VC separator. When the switch is in a congested state, it reduces the ER field of each passing backward RM cells to MACR$*$ERF if ACR is larger than MACR$*$DPF. The ER of the VCs whose ACR is less than MACR$*$DPF need not be reduced in order to keep the fairer behaviour. This manner is known as intelligent marking. When the switch is in a very congested state, it reduces ER to MACR$*$MRF.

## 3. Simulation model and parameters

### *3.1. Simulation model*

The simulation model is depicted in Figure 2. There are 10 unidirectional connections with source *i* sending data to destination *i* through the switch. Each source and destination has three components: TCP, IP, AAL and ATM. The user data have infinite backlog, i.e. there are always data to transmit. The one-way propagation delay is denoted by $\tau$. The buffer service policy at the switch is a FIFO.

We implement a TCP version with fast retransmission, but no fast recovery. Also we do not implement the EPD (early packet discard) and PPD (partial packet discard). In other words, the switch drops individual cells, rather than whole and partial packets. In ABR flow control, EPRCA algorithm is used at the switch in our simulation.

The bandwidth of the link between two switches is 365566 cells/s, i.e. 155 Mbps. The bottleneck is caused by the link being shared by 10 sources. Therefore, it is when some cells are queueing in the buffer of switch 1 that congestion occurs, whereas switch 2 does not become a bottleneck at any time.

If there is no cell loss, the system can transmit 365566 cells/s ideally including RM cells and data cells from sources to destinations. From the TCP layer's view, it can transmit

$$365566 \times 48 \times \frac{31}{32} \times \frac{\mathrm{MSS}}{\mathrm{MSS} + 40} \text{ bytes/s}$$

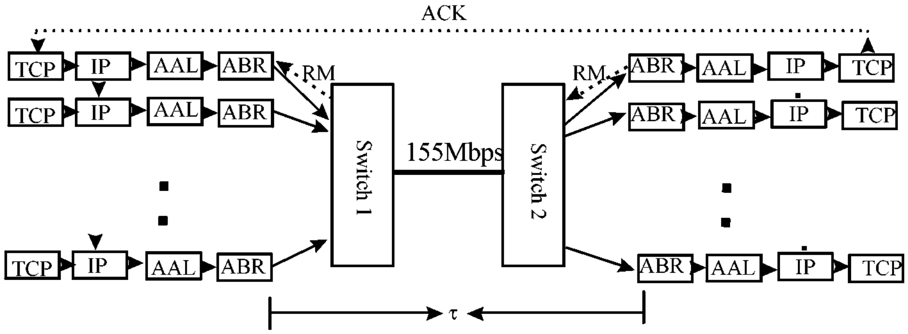


Figure 2. Simulation model

Table I. Parameters of simulation

| Protocol | Parameter | value |
|---|---|---|
| TCP | MSS (maximum segement size) | 9148 bytes |
| TCP | Wrcv (receiver buffer size) | 64036 bytes |
| TCP | Grain (lock granularity) | 0·1 s |
| ABR | PCR (peak cell rate) | 365566 cells/sec |
| ABR | MCR (minimum cell rate) | 0 |
| ABR | ICR (initial cell rate) | PCR/20 |
| ABR | Nrm | 32 |
| ABR | RDF (rate decrease factor) | 1/16 |
| ABR | RIF (rate Increase factor) | 1/16 |
| EPRCA | Q (switch buffer size) | 2000 cells |
| EPRCA | IMR (initial rate for MACR) | PCR/100 |
| EPRCA | AV (exponential averaging factor) | 1/16 |
| EPRCA | VCS (VC separator) | 7/8 |
| EPRCA | ERF (explicit reduction factor) | 15/16 |
| EPRCA | DPF (down pressure factor) | 7/8 |
| EPRCA | MRF (major reduction factor) | 1/4 |

Taking MSS equal to 9148 for example, the TCP effective throughput is

$$365566 \times 48 \times \frac{31}{32} \times \frac{9148}{9148 + 40} = 1{\cdot}692 \times 10^{7} \text{ bytes/s}$$

### *3.2. Parameters*

Some parameters used in the experiments are listed in Table I. Other parameters, which are used in the ABR rate-based control, have the default values defined in ATM Forum 4.0.[16] Note that the value of RDF has no influence on performance in our model because the EPRCA switch does not set the CI bit of the passing RM cells.

Copyright © 1999 John Wiley & Sons, Ltd.

Table II. Parameters of cell-loss-free and cell-loss cases

| Cell-loss-free case | | Cell-loss case | |
|---|---|---|---|
| Parameters | Value | Parameters | Value |
| $Q_L$ | 500 cells | $Q_L$ | 800 cells |
| DQT | 800 cells | DQT | 1500 cells |
| $\tau$ | 0·01 ms | $\tau$ | 1 ms |

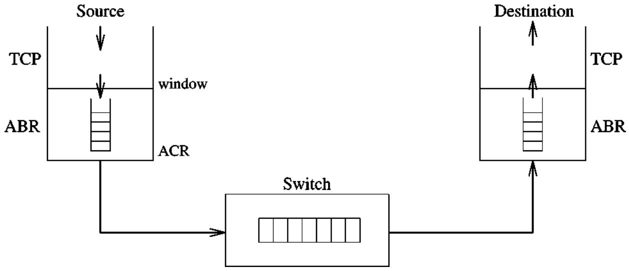


Figure 3. View of TCP over ABR

### *3.3. Cell-loss-free and cell-loss cases*

Two cases are distinguished to show the effect of the dual control. One is cell-loss-free case, the other is cell-loss case. Table II shows the parameters in both cases.

## 4. Effects and suggestions

### *4.1. Interaction of TCP and ABR*

*4.1.1. Window-based vs. rate-based.* We can view the unidirectional data traffic of TCP over ABR as shown in Figure 3. The TCP end system sends packets to the ABR end system. The amount of packets sent depends on the TCP window. The ABR end system sends cells (divided packets) to the switch at ACR. The switch switches cells at a constant rate. When the TCP end system sends faster than the ABR end system, there are cells queued in the ABR end system. In such a period, ABR flow control dominates the sending rate of the combined system, which is called rate-based. When the TCP end system sends slower than the ABR end system, the queue of ABR end system is always empty. TCP flow control dominates in this case, which is called window-based. The window-based period appears when TCP window is small or propagation delay is large. This is because TCP stops to wait for the acknowledgement after a 'window' of data is transmitted.

In the cell-loss-free case, the performance is always rate-based except at the beginning of a connection when the TCP window is small but is increasing quickly. In the cell-loss case, the

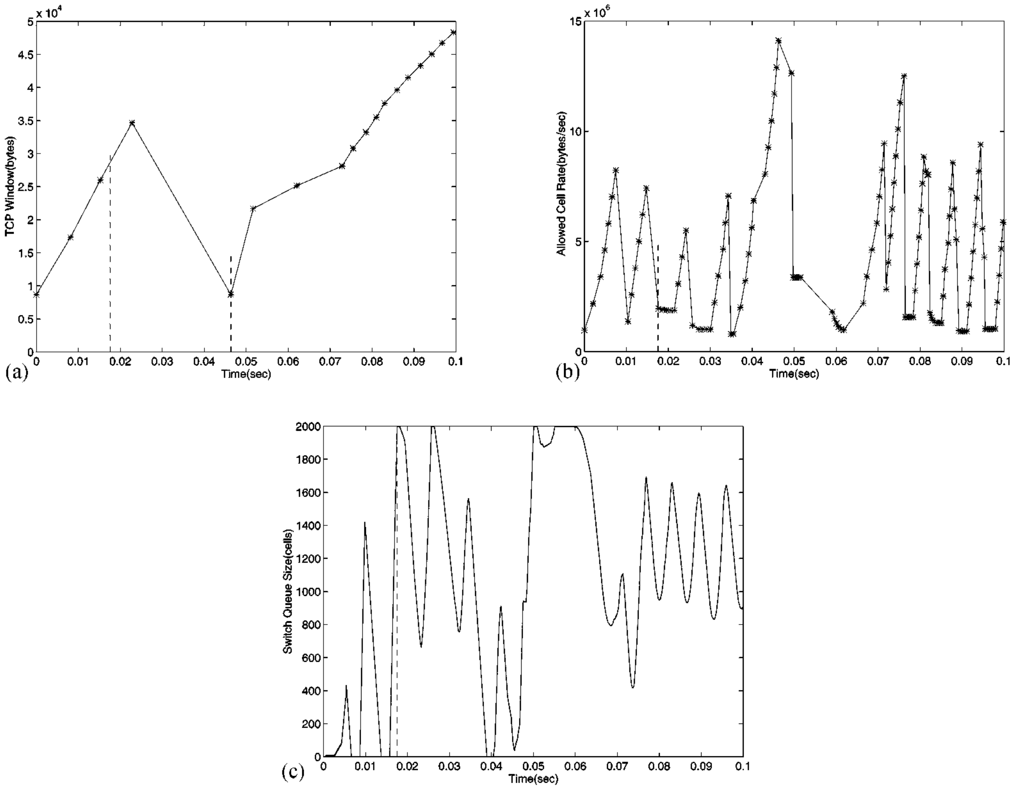


Figure 4. Time-dependent behaviour of TCP over ABR. (a) TCP window of SES 1. (b) ACR of SES 1. (c) Switch queue size

system alternates between window- and rate-based periods because the TCP window size drops when packet loss occurs.

*4.1.2. Asynchronous response of TCP and ABR.* Figure 4 shows the time-dependent behaviour of our simulation. Comparing Figure 4(a) with 4(b), we observe that ACR changes more often than the TCP window. As we know, the ABR end system adjusts ACR when the RM cell returns and the TCP end system changes the TCP window when the acknowledgment is received. Since one packet is divided into more than Nrm cells, ACR changes more often.

Furthermore, the response to packet loss is asynchronous. When congestion occurs due to cell loss, ABR flow control decreases its sending rate suddenly to solve the congestion. When TCP flow control starts its congestion control, the congestion might be relieved already. Obviously, they cannot co-operate well to solve the congestion and TCP flow control even causes further unnecessary performance degradation. The congestion shall be resolved solely by ABR flow control due to the delayed reaction of TCP layer.

Copyright © 1999 John Wiley & Sons, Ltd.

In summary, we say that TCP flow control and ABR flow control cannot co-operate well because (1) the combined sending rate alternates between rate-based or window-based, i.e. the dual control cannot behave better than the single control, and (2) the adjustment frequency and the response to congestion are asynchronous.

### *4.2. Phenomena and solutions*

*4.2.1. Unused high ACR and underflowed switch queue.* We conducted two simulation experiments for cell-loss-free and cell-loss networks to investigate the dual control. Figures 5 and 6 show TCP window and ACR behaviour of one connection and switch queue behaviour. Since ABR flow control is fair, one connection can represent other connections.

Comparing ABR behaviour in Figures 5(b) and 6(b), in Figure 5(b), ACR oscillates between $0{\cdot}4 \times 10^6$ to $4 \times 10^6$ bytes/s, but in Figure 6(b), there are some circumstances that cause ACR to be

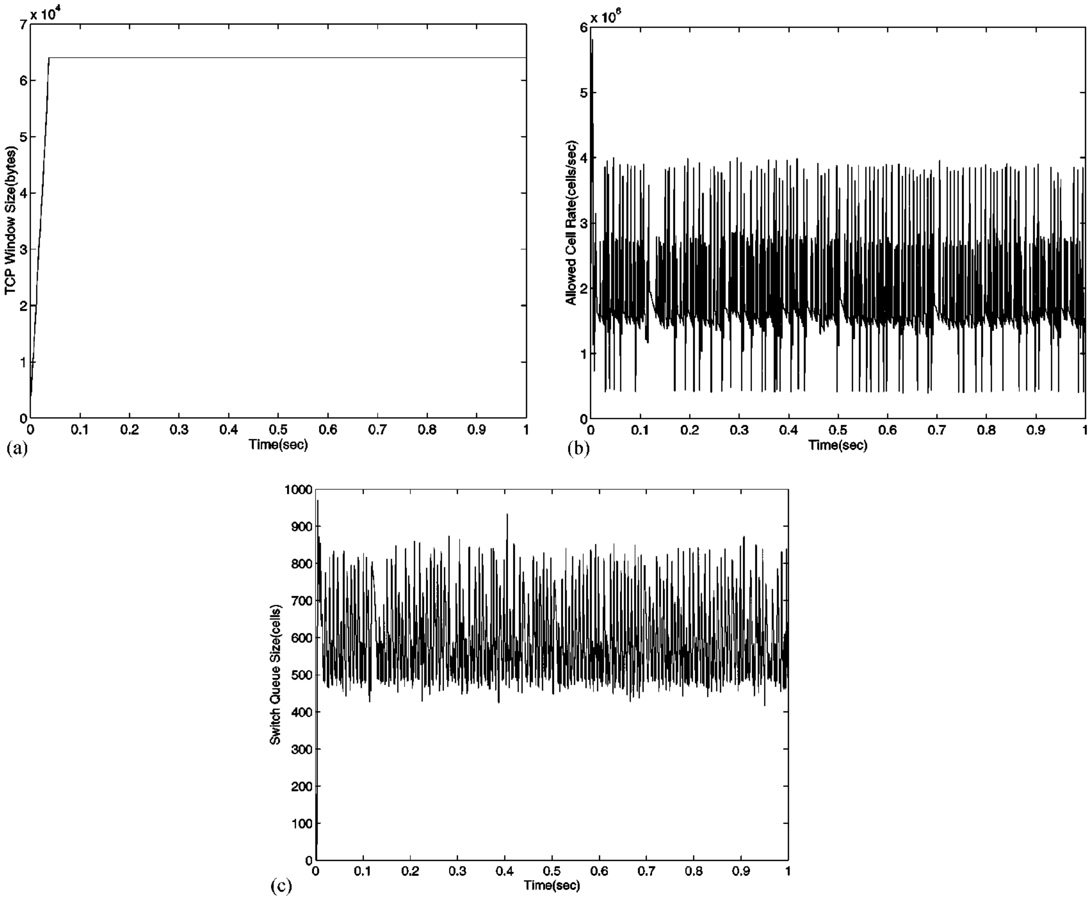


Figure 5. Time-dependent behaviour in cell-loss-free case. (a) TCP window of SES 1. (b) ACR of SES 1. (c) Switch queue size

Copyright © 1999 John Wiley & Sons, Ltd.

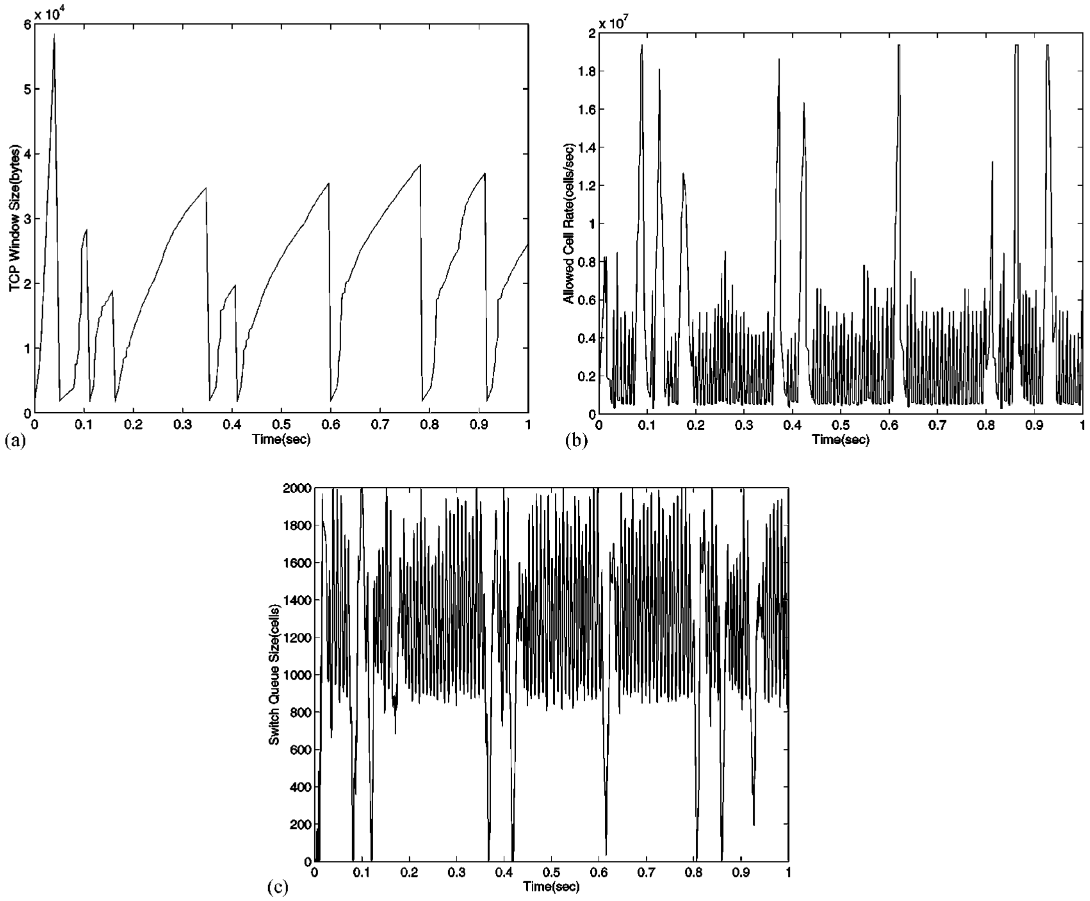


Figure 6. Time-dependent behaviour in cell-loss case. (a) TCP window of SES 1. (b) ACR of SES 1. (c) Switch queue size

very high. If any cells of a packet are lost, the destination cannot assemble the packet successfully. The lost packet is detected after receipt of three duplicate acknowledgements for fast retransmission and the TCP window is set to one packet. The drop of a TCP window and the succeeding slow start make the switch queue shrink. Hence ACR is increased to a much higher value. The high ACR is not fully used, but when the traffic from the TCP layer to ATM grows later on, the high ACR will lead cells to swamp the switch buffer. Cells may be lost and it is much worse in the configurations with a large number of connections. The high ACR should be reclaimed. The reclamation of unused bandwidth is the so-called *use-it-or-lose-it policy* in TM4.0.[16] It is optionally implemented. Because a single cell-loss means an effective packet loss, TCP performs often a slow start, especially in congested networks. Therefore, it is important to implement the use-it-or-lose-it policy in ABR flow control.

Second, we compare the switch behaviour in Figures 5(c) and 6(c). When the packet gets lost and slow start is performed, the switch queue underflows. Underflow of the switch queue leads to

Copyright © 1999 John Wiley & Sons, Ltd.

lower throughput. This problem is also discovered in TCP over the packet network. The TCP Reno version solves it with the addition of *fast recovery* that sets *cwnd* to half of the TCP window and performs congestion avoidance, instead of slow start, when congestion occurs.[18] If fast recovery is added, the chance of having switch queue underflow as well as unused high ACR can be lowered.

## 5. Conclusion

In this paper, we investigate TCP flow control over ABR flow control with the ATM EPRCA switch. We summarize and list the results:

1. TCP flow control cannot co-operate with ABR flow control well.
2. When a packet is lost, the interaction of TCP flow control and ABR flow control may cause the unused high ACR and switch queue underflow. We suggest to implement the use-it-or-lose-it policy in ABR flow control and fast recovery in TCP flow control to alleviate these problems.

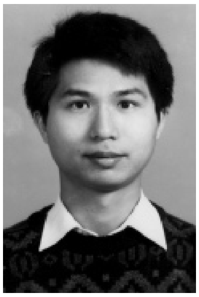
In the future, some issues will also be of our concerns. The use-it-or-lose-it policy should be implemented according to the characteristics of TCP flow control and ABR flow control. Also, we have pointed out that fast recovery can solve the switch queue underflow and unused high ACR. It is necessary to investigate the amount of improvement. Finally, some methods for improving performance of TCP over ABR, e.g. EPD and PPD, should be further studied in the future.

## References

1. Internet 2 General Information, http://www.internet2.edu.
2. R. J. Gurski and C. L. Williamson, 'TCP over ATM: simulation model and performance results', *Proc. IEEE ICC'96*, pp. 328–335, March 1996.
3. A. Bianco, 'Performance of the TCP protocol over ATM network', *Proc. ICCCN'94*, pp. 170–177, September 1994.
4. M. Hassan, 'Impact of cell-loss on the efficiency of TCP/IP over ATM', *Proc. ICCCN'94*, pp. 165–169, September 1994.
5. K. Moldklev and P. Guningberg, 'How a large ATM MTU causes deadlocks in TCP data transfers', *IEEE/ACM Trans. Networking*, **3**(4), 409–422 (1995).
6. A. Romanow and S. Floyd, 'Dynamics of TCP traffic over ATM networks', *IEEE JSAC*, **13**(4), (1995).
7. C. Tipper and J. Daigle, 'ATM cell delay and loss for best-effort TCP in the presence of isochronous traffic', *IEEE JSAC*, **3**(8), 1457–1464 (1995).
8. H. Li, K.-Y. Siu, H.-Y. Tzeng, C. Ikeda and H. Suzuki, 'A simulation study of TCP performance in ATM networks with ABR and UBR services', *Proc. IEEE INFOCOM'96*, Vol. 3, pp. 1269–1276, March 1996.
9. G. Hasegawa, H. Ohsaki, M. Murata and H. Miyahara, 'Performance evaluation and parameter tuning of TCP over ABR service in ATM networks', *IEICE Trans. Commun.*, **E79-B** (5), (1996).
10. H. Saito, K. Kawashima, H. Kitazume, A. Koike, M. Ishizuka and A. Abe, 'Performance issues in public ABR service', *IEEE Commun. Mag.* (1996).
11. S. Kalyanaraman, R. Jain, S. Fahmy, R. Goyal, F. Lu and S. Srinidhi, 'Performance of TCP/IP over ABR', *Globecom'96*, November 1996.
12. C. Fang and H. Chen, 'TCP performance simulations of enhanced PRCA scheme', ATM Forum 94-0932, September 1994.
13. D. Sisalem, 'Rate based congestion control and its effects on TCP over ATM', http://ptolemy.eecs.berkeley.edu/papers/tcpSim.
14. S. Floyd, 'TCP and explicit congestion notification', ftp://ftp.ee.lbl.gov/papers/tcp_ecn.4.ps.Z, 1994.
15. P. Calhoun, 'Congestion control in IPv6 internetworks', Internet draft, May 1995.
16. The ATM Forum, 'Traffic management specification version 4.0', ftp://ftp.atmforum.com/pub/approved-specs/af-tm-0055.000.ps, April 1996.
17. V. Jacobson, 'Congestion avoidance and control', *Proc. ACM SIGCOMM'88*, August 1988.
18. V. Jacobson, 'Berkeley TCP evolution from 4.3-Tahoe to 4.3-Reno', *Proc. Eighteenth Internet Engineering Task Force*, pp. 363–366, September 1990.

Copyright © 1999 John Wiley & Sons, Ltd.

**Authors' biographies:**



**Yuan-Cheng Lai** received the BS and MS degrees in Computer Science and Information Engineering from National Taiwan University in 1988 and 1990, respectively. He received his PhD degree from the Department of computer and Information Science of National Chiao Tung University in 1997. From 1992 to 1994, he was an associate researcher at Computer & Communication Research Labs of Industrial Technology Research Institute, Taiwan. He joined the faculty of the Department of Computer and Information Science at National Cheng Kung University in August 1998 and is now Assistant Professor. He can be contacted at laiyc@locust.csie. ncku.edu.tw.

**Ying-Dar Lin** received the Bachelor's degree in Computer Science and Information Engineering from National Taiwan University in 1988, and the MS and PhD degrees in Computer Science from the University of Californuia, Los Angeles in 1990 and 1993, respectively. He joined the faculty of the Department of Computer and Information Science at National Chiao Tung University in August 1993 and is now Associate Professor. His research interests include design and analysis of high-speed LANs/MANs/WANs, high-speed switching and routing, and network-centric computing. Dr Lin is a member of ACM and IEEE. He can be contacted at ydlin@cis.nctu.edu.tw.

**Hsiu-Fen Hung** received the BS degree in Information and Computer Education from National Taiwan Normal University in 1995, and MS degree in Computer and Information Science form National Chiao Tung University in 1997. She is now an assistant engineer in Chunghwa Telecom Labs.

Copyright © 1999 John Wiley & Sons, Ltd.