TCP 的交通控制在 ABR 的流量控制之上的問題研究

The Problem of TCP over ATM EPRCA Rate-based Flow Control

賴源正　林盈達　洪秀芬
Yuan-Cheng Lai　Ying-Dar Lin　Hsiu-Fen Hung

交通大學資訊科學研究所
Department of Computer and Information Science
National Chiao Tung University

摘要

在這篇論文中，我們模擬並審視雙層控制，也就是傳輸控制協定層的交通控制及非同步傳輸網路層的可利用位元速率式服務流量控制。首先,我們觀察到傳輸控制協定中的交通控制及可利用位元速率式流量控制並不能配合得很好。更糟的是在封包遺失之後所啓動的慢起動會造成很高卻沒有使用的允許細胞速率及交換器佇列下溢。我們建議要實作使用或遺失政策和快速復原來解決這二個問題。

關鍵字：可利用位元速率，流速控制，雙層控制，使用或遺失。

Abstract

In this paper, we investigate the dual control—TCP flow control at TCP layer and ABR flow control at ATM layer. First, we observe that TCP flow control and ABR flow control cannot cooperate 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 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.

Keywords: ABR, flow control, dual control, use-it-or-lose-it.

1. Introduction

ATM (Asynchronous transfer mode) 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 Fig. 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 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.

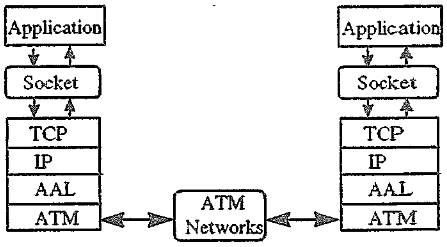


Figure 1. TCP/ATM Protocol Stack.

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 service [2-7]. It is largely due to 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 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].

TCP achieves better performance with ABR service than with UBR service plus early packet discard scheme in some cases [8-10]. ABR service provides fair bandwidth allocation and high link utilization and requires relatively small switch buffer in a LAN environment [8]. In a WAN environment with 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 [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 study, we investigate the time dependent behavior 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 acknowledgment is received, the TCP end system sets TCP window as the minimum of the advertisement window and the congestion window(*cwnd*). The advertisement window specifies 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 acknowledgment is received. This is the slow start algorithm. After TCP window is larger than *ssthresh* (a slow start threshold), congestion avoidance process is performed, where *cwnd* is only increased by 1/*cwnd* packet each time. *ssthresh* is initialized to 65536 bytes. When a packet is lost, one-half of the current TCP window is saved in *ssthresh*, and 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 to 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 [17].

In addition to the expiration of the retransmission timer, the duplicate acknowledgments can be used to detect the loss of a packet. When three or more duplicate acks 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 [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 utilization.
- Clock granularity (Grain) : Current TCP algorithm uses a clock granularity of 300-500ms to measure the round-trip-time. It's too course in a high-speed low propagation delay ATM environment. Allyn Romanow suggested to set it to 0.1ms [6], but Kalyaanaraman suggested 100ms [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 as 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 fields in RM cells, the new ACR is computed as

$$\begin{aligned} &ACR=\max(\min(ACR+RIF*PCR, ER), MCR), && \text{if } CI=0,\\ &ACR=\max(\min(ACR*(1-RDF), ER), MCR), && \text{if } CI=1, \end{aligned}$$

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_L) and the very congested threshold(DQT) to determine state of the network. If the queue size exceeds Q_L, it is in a congested state. If the queue size exceeds DQT, it is in a very congested state. 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 MACR=MACR+(ACR-MACR)*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 ER field of each passing backward RM cells to the minimum of MACR*ERF and ER if ACR is larger than MACR*DPF (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 Fig. 2. There are ten 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 has infinite backlog, i.e. there is always data to transmit. The one-way propagation delay is denoted by τ. The buffer service policy at the switch is a FIFO.

We implement a Tahoe TCP version without fast retransmission and no fast recovery. Also we do not implement the EPD (Early packet discard) and PPD (partial packet). In other words, the switch drops individual cells, rather than whole and partial packets. In ABR flow control, ERICA algorithm is used at the switch in our simulation.

The bandwidth of the link between two switches is 365566 cells/sec, i.e. 155Mbps. The bottleneck is the link shared by ten sources. Therefore, some cells are queueing in the buffer of switch 1 to cause congestion. Meanwhile, switch 2 do not become a bottleneck at any time.

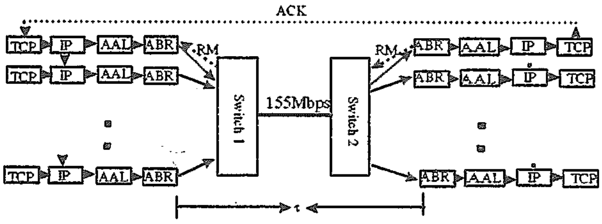


Figure 2. Simulation model.

3.2 Parameters

Some parameters used in the experiments are listed in table 1. 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 to performance in our model because EPRCA switch does not set CI bit of the passing RM cells.

Protocol	Parameter	value
TCP	MSS (Maximum segemnet size)	9148
TCP	Wrcv (Receiver buffer size)	64036
TCP	Grain (lock granularity)	0.1 s
ABR	PCR (Peak cell rate)	365566
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
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

Table 1. Parameters of simulation.

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 2 shows the parameters in both cases.

Cell-loss-free case		Cell-loss case	
Parameters	Value	Parameters	Value
Q_L	500 cells	Q_L	800 cells
DQT	800 cells	DQT	1300 cells
τ	0.01 ms	τ	1 ms

Table 2. Parameters of cell-loss-free and cell-loss cases.

4. Effects and Suggestions

Window-based vs. rate-based

We can view the unidirectional data traffic of TCP over ABR as shown in Fig. 3. The TCP end system sends packet to the ABR end system. The amount of packets sent depends on TCP window. The ABR end system sends cells (divided packets) to the switch at ACR. The switch switches cells at the 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 acknowledgment after a "window" of data are transmitted.

In the cell-loss-free case, the performance is always rate-based except the beginning of a connection where the TCP window is small but is increasing quickly. In the cell-loss case, the system alternates between window-based and rate-based periods because the TCP window size drops when packet loss occurs. The observation of alternate window-based and rate-based periods is important in analyzing the simulation results of section 5.

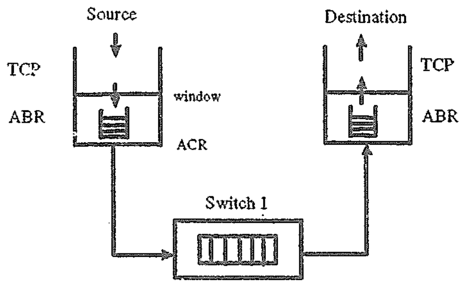
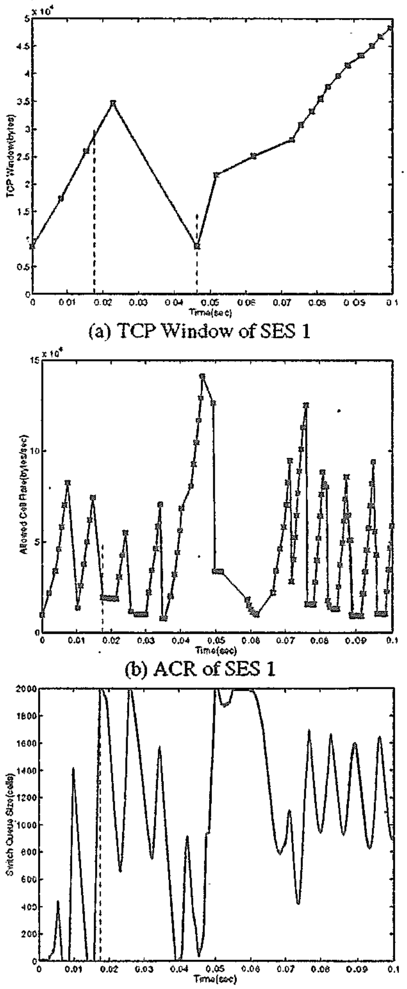


Figure 3. View of TCP over ABR.

Asynchronous response of TCP and ABR

Fig. 4 shows the time dependent behavior of our simulation. Comparing Fig. 4(a) with Fig. 4(b), we observe that ACR changes more often than TCP window. As we know, the ABR end system adjusts ACR when the RM cell returns and the TCP end system changes 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 the 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 cooperate well to solve the congestion and TCP flow control even causes the unnecessary performance degradation. The congestion shall be resolved solely by ABR flow control due to the delayed reaction of TCP layer.



(a) TCP Window of SES 1

(b) ACR of SES 1

(c) Switch Queue Size

Figure 4. Time dependent behavior of TCP over ABR.

In summary, we say that TCP flow control and ABR flow control cannot cooperate 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.

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. Fig. 5 and 6 show TCP window and ACR behavior of one connection and switch queue behavior. Since ABR flow control is fair, one connection can represent other connections.

Comparing ABR behavior in Fig. 5(b) and Fig. 6(b), in Fig. 5(b), ACR oscillates between $0.4*10^6$ to $4*10^6$ bytes/sec, but in Fig. 6(b), there are some circumstances that ACR is 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 acknowledgments for fast retransmission and TCP window is set to one packet. The drop of 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 TCP layer to ATM grows later on, the high ACR will lead cells to swamp the switch buffer. Cells may lose. 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 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 slow start often, especially in the congested networks. Therefore, it is important to implement use-it-or-lose-it policy in ABR flow control.

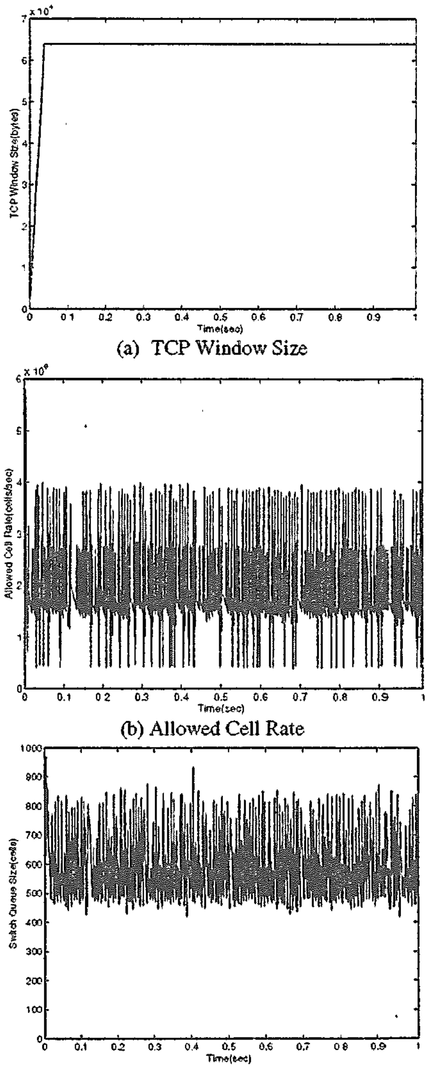
Second, we compare the switch behavior on Fig. 5(c) and Fig. 6(c). When the packet gets lost and slow start is performed, the switch queue underflows. Underflow of the switch queue leads to lower throughput. This problem is also discovered in TCP over the packet network. TCP Reno version solves it with 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 work, 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 cooperate 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 are also our concerns. The use-it-or-lose-it policy should be implemented according to the characteristics of TCP flow control and ABR flow control. We have pointed that fast recovery can solve the switch queue underflow and unused high ACR. It is necessary to investigate the amount of improvement. Also, other switch mechanisms can be considered in the future study.

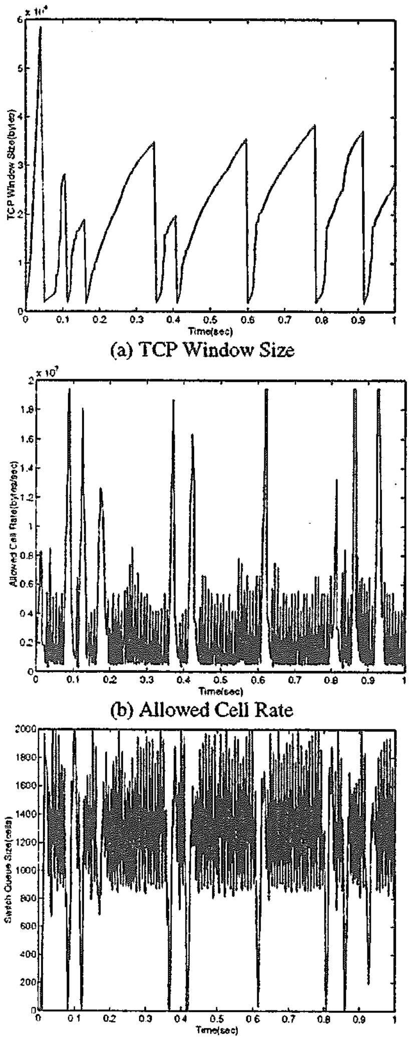


(a) TCP Window Size

(b) Allowed Cell Rate

(c) Switch Queue Size

Figure 5. Time dependent behavior in cell-loss-free case.



(a) TCP Window Size

(b) Allowed Cell Rate

(c) Switch Queue Size

Figure 6. Time dependent behavior in cell-loss case.

Reference

[1]Internet 2 General Information, http://www.internet2.edu.

[2]Gurski, R.J. and Williamson, C.L., "TCP over ATM : Simulation Model and Performance Results", Conference Proceedings of the 1996 IEEE fifteenth Annual International Phoneix Conference on Computers and Communications, pp328-335, March 1996.

[3]A. Bianco, "Performance of the TCP Protocol over ATM Network", Proceedings of the 3rd International Conference on Computer Communications and Networks, pp170-177, September 1994.

[4]M. Hassan, "Impact of Cell-loss on the Efficiency of TCP/IP over ATM", Proceedings of the 3rd International Conference on Computer Communications and Networks, pp165-169, September 1994.

[5]K. Moldklev and P. Guningberg, "How a large ATM MTU Causes Deadlocks in TCP Data Transfers", IEEE/ACM Transactions on Networking, Vol. 3, No.4, pp409-422, August 1995.

[6]Allyn Romanow and Sally Floyd, "Dynamics of TCP Traffic over ATM Networks," IEEE Journal on Selected Areas in Communications, VOL.13, NO.4, May 1995.

[7]C. Tipper and J. Daigle, "ATM Cell Delay and Loss for Best-Effort TCP in the Presence of Isochronous Traffic", IEEE Journal on Selected Areas in Communications, Vol. 3, No.8, pp1457-1464, October 1995.

[8]Hongqing Li, Kai-Yeung Siu, Hong-Yi Tzeng, Chinatsu Ikeda, and Hiroshi Suzuki , "A Simulation Study of TCP Performance in ATM Networks with ABR and UBR services", Proceedings of IEEE INFOCOM'96 Conference on Computer Communications, Vol.3, pp1269-1276, March 1996.

[9]Go Hasegawa, Hiroyuki Ohsaki, Masayuki Murata, and Hideo Miyahara, "Performance Evaluation and Parameter Tuning of TCP over ABR Service in ATM Networks," IEICE TRANS. COMMUN., VOL .E79-B, NO.5 MAY 1996.

[10]Hiroshi Saito, Konosuke Kawashima, Hideo Kitazume, Arata Koike, Mika Ishizuka, and Atsushi Abe, "Performance Issues in Public ABR Service," IEEE Communications Magazine, Nov 1996.

[11]Shiv Kalyanaraman, Raj Jain, Sonia Fahmy, Rohit Goyal, Fang Lu, Saragur Srinidhi, "Performance of TCP/IP over ABR," Proceedings of Globecom'96,November 1996.

[12]C. Fang and H. Chen, "TCP performance simulations of enhanced PRCA scheme," ATM Forum 94-0932, September 1994.

[13]Dorgham 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," in Proceeding of SIGCOMM '88, ACM, Aug. 1988.

[18]V. Jacobson, "Berkeley TCP evolution from 4.3-Tahoe to 4.3-Reno," Proceedings of the Eightheenth Internet Engineering Task Force, pp. 363-366, September 1990.