Proceedings of **ITC-CSCC 2000**, Pusan, Korea

Congestion Control in TCP over ATM –UBR Networks

Woo Chool Park, Sang Jun Park and Byung Ho Rhee
Department of Electronics Engineering, Hanyang University
1271 Sa-Dong ,Ansan, Kyunggi, 425-791, KOREA
Tel: +82-345-400-4045, Fax: +82-345-416-8847
E-mail: wcpark, parksang, bhrhee@hymail.hanyang.ac.kr

Abstract

In this paper we approach the problem of congestion control for TCP traffic over ATM-UBR networks by focusing on the fact that to get best performance. We study how to efficiently support TCP traffic in the subnet ATM model, when ATM is only a single link in the whole path. We show that when UBR connection.

We analyze the ATM-UBR network service using the BSD 4.3 Reno, Tahoe TCP. However we found the fact that the characteristic of fast recovery algorithm makes a serious degradation of performance in multiple cell loss drop situation. We propose new fast recovery algorithm to solve the problem.

1. Introduction

ATM is expected to be widely used in networks. High speed, integrated service support, and scalability all contribute to its popularity. At the same time, the TCP/IP protocol suite is expected to become ubiquitous, as the popularity of the Internet, corporate intranets, and new applications such as the internet phone and groupware tools spreads.

Though building and managing future integrated networks would be much easier if either one of these protocol suites were to become the single universal networking standard, most researchers expect that they will coexist. Consequently methods for their integration will be needed. In the model the ATM connection is only a single link in the end-to-end TCP connection. We call the former the end-to-end ATM model that is called the subnet ATM model.

In this paper we are concerned with the performance of TCP connections over ATM networks. Many factors influence the performance of TCP over ATM. Among them significant ones are the network model relating TCP connections to ATM connections, the type of ATM switches, and the characteristics of the congestion control mechanisms of TCP itself.[1,2]

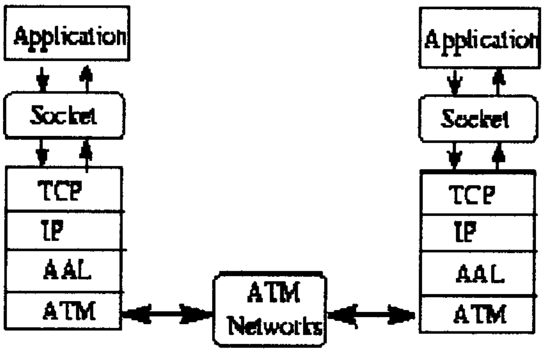

Fig 1. TCP over ATM protocol stack

We analyze the ATM-UBR network service using

We analyze the ATM-UBR network service using the BSD 4.3 Reno, Tahoe TCP. However we found the fact that the characteristic of fast recovery algorithm makes a serious degradation of performance in multiple cell loss drop situation. We propose new fast recovery algorithm to solve the problem.[3]

2. The new fast recovery algorithm

We derive formulas for the average throughput during a congestion avoidance cycle by a TCP Reno connection served at a fixed rate. We assume that a single TCP Reno connection is supported by the rate of μ. The connection's RTT is τ, and the available buffer space is B.

A TCP Reno source first goes through the slow start cycle and then continuously repeats a cycle of congestion avoidance followed by a packet drop. The TCP connection's window size is at its maximum when the packet drop occur, the TCP Reno source enters the congestion avoidance phase again. At any point of time, the current window size, cwnd, controls the number of packets in pipe between the source and the destination. As maximum number of packets that can be in the pipe is the sum of the bandwidth-delay product and buffer size, packet loss occurs when the window size is greater than the sum of the buffer size and the bandwidth-delay product, cwnd > B + $\mu\tau$.

Consequently the maximum window size is $W_M = \mu\tau$. if the maximum window size is W_M, the size of the slow-start threshold $W_{SS} = \frac{W_M}{2}$. During the congestion avoidance phase, the window opens linearly in increments of one each round trip time, so the number of round trips until the window increases from W_{SS} to W_M is equal to $W_M - W_{SS} = W_{SS}$.

The average long term throughput(λ) of the TCP connection can be calculated from the congestion avoidance cycle by the following equation,

$$\lambda = \frac{N_{CA}}{t_{CA}} \qquad (1)$$

where N_{CA} is the number of packets transmitted during the congestion avoidance period and t_{CA} is the duration of the congestion avoidance period.[4,5]

We can divide t_{CA} into t_{CA1}, denoting the time until the window size equals the bandwidth-delay product, and t_{CA2} the time from the end of t_{CA1} until packet loss actually occurs. The t_{CA} is consisted of t_{CA1} + t_{CA2}.

$$t_{CA} = \frac{(W_M+1)\tau}{2} + \frac{(W_M-\mu\tau)(W_M-\mu\tau+2)}{2\mu} \qquad (2)$$

Since during congestion avoidance the number of packets transmitted in each round trip time increases linearly from W_{SS} to W_{CA} in increments of one for each round trip time. We can calculate N_{CA} as follows.

$$\begin{aligned} N_{CA} &= W_{ss}(W_M - W_{ss}+1) + \frac{(W_M - W_{ss})^2}{2} \\ &= \frac{3W_M^2}{8} + \frac{W_M}{2} \\ &= \frac{3(B+\mu\tau)^2}{8} + \frac{(B+\mu\tau)}{2} \qquad (3) \end{aligned}$$

Relating (2) and (3) to (1) we get

$$\lambda = \frac{\frac{3}{8}\left(\frac{W_M}{\mu\tau}\right)^2 + \frac{1}{2}\frac{W_M}{\mu\tau}\frac{1}{\mu\tau}}{\frac{1}{2\mu}\left(\frac{W_M}{\mu\tau}-1\right)^2 + \frac{1}{\mu}\left(\frac{W_M}{\mu\tau}-1\right)\frac{1}{\mu\tau} + \left(\frac{1}{2}\frac{W_M}{\mu\tau}+\frac{1}{\mu\tau}\right)\frac{1}{\mu}} \qquad (4)$$

The parameter of τ , μ is constant, we are interesting the parameter which is W_M.

We use the W_M parameter to control the congestion control algorithm.[6,7]

We propose the new fast recovery algorithm which control the congestion windows size.[8]

when the third duplicate ACK is received,
step 1. set ssthresh value equals
ssthresh =max(flightSize/2, 3*MSS)
step 2. Retransmit the lost segment.
step 3. set cwnd to ssthresh.
step 4. For each additional duplicate ACK received, increment cwnd by MSS.

Table 1. The proposed TCP congestion control algorithm.

3. The simulation model

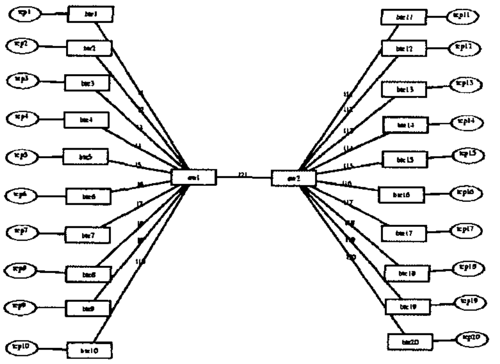

Fig. 1 The Model of Simulation

In this paper, we use the EPD buffering algorithm, and NIST ATM switch algorithm.[9] The link between two switch is 155Mbps, and 10 sources, destinations. This network has the bottleneck problem.

프로토콜	parameter	value
TCP	MSS(Maximum segment size)	9160byte
TCP	Wrecv(Receiver buffer size)	64000 byte
TCP	Grain(lock granularity)	100ms
nist switch	Queue(switch buffer size)	2000cells
nist switch	buffer algorithm	EPD
l1-120: 0.1Km, 121 : 0.3Km		

Table 2. The Parameters of Simulation

The Table 2 is the parameter value used in this paper simulation. The value of parameter has referenced ATM Forum4.0.[10]

4. The Performance Analysis

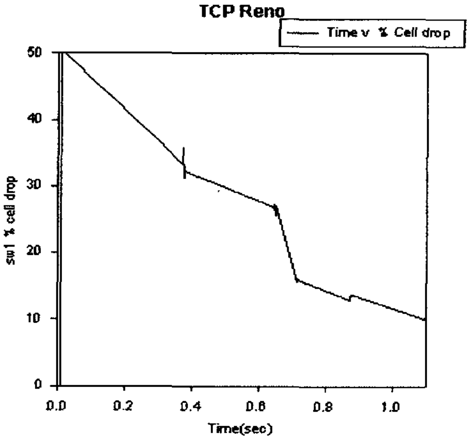

Fig 2. % cell drop in SW1 using Reno TCP

The BSD 4.3 Reno, TCP congestion control algorithms: Slow Start, congestion Avoidance, Fast Retransmit, Fast Recovery.

The 4.3 BSD Reno TCP is currently

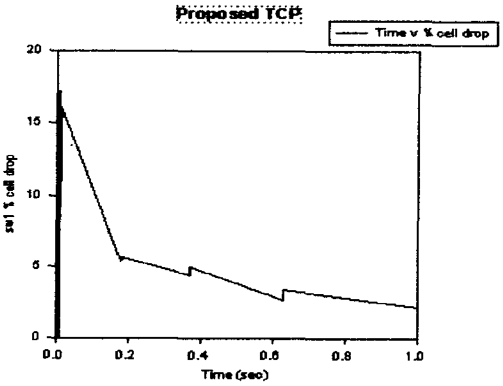

Fig 3. % cell drop in SW1 using Proposed TCP

used in the world. The figure 4 reveal the cell drop rate in the simulation model.

The following figure is new proposed TCP congestion algorithm to improve the problem in increasing cell loss rate in Fast Recovery. According to the equation (4) We proposed the new control W_M to the performance improvement in average throughput.

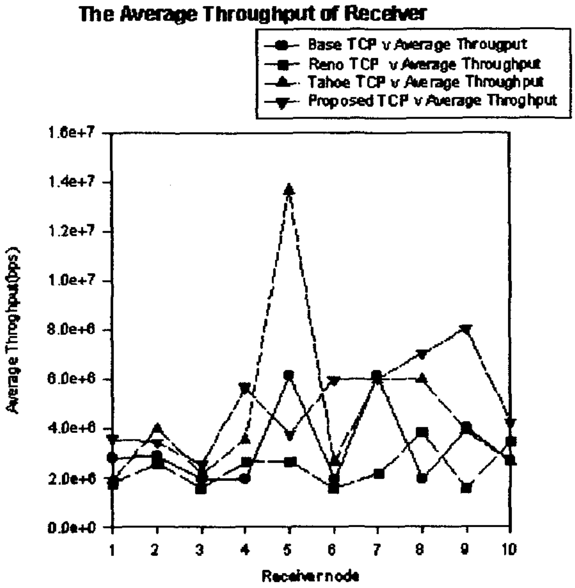

Fig 4. The Average Throughput of Receiver

5. Conclusion

In this paper, we analyze TCP congestion control algorithm in ATM-UBR networks. TCP congestion control algorithm is consist of Slow Start, Congestion Avoidance, Fast Recovery, Fast Retransmit.

We are currently using 4.3 BSD Reno TCP. However, the 4.3 BSD Reno TCP has the problem of performance degradation in multiple cell loss. we have proposed the new fast recovery algorithm when multiple cell loss occurs in ATM-UBR networks.

References

[1] T.V.Lakshman, Arnold Neidhardt, and Tenuis J. Ott, "The Drop from Front Strategy in TCP and in TCP over ATM", IEEE INFOCOM'96 pp. 1242 -1250 Mar. 1996.

[2] Romanow, A. and Floyd, S., "Dynamics of TCP traffic over ATM Networks", ACM SIGCOMM Computer Communications Review, October 1994.

[3]Fall, K and S. Floyd, "Simulation -based Comparisons of Tahoe, Reno and SACK TCP",Computer Communication Review, July 1996.

[4]Stevens, W., "TCP Slow Start, Congestion Avoidance,FastRetransmit and Fast Recovery Algorithms", RFC 2001, January 1997.

[5]R. Stevens, TCP/IP Illustrated: The Protocols, vol. 1.ISBN-0201-63346-9: Addison-Wesley, 1994

[6] M. Allman, V. Paxson, W. Stevens, "TCP Congestion Control", RFC 2581, April 1999

[7] S. Floyd and T. Henderson, "The NewReno Modification to TCP's Fast Recovery Algorithm", RFC 2582, April 1999

[8] The UCB/LBNL/VINT Network Simulator (NS). URL "http://www -mash.cs.berkeley.edu/ns/".

[9] Golmie,N., Chang, Y.,Su, D., "NIST ER Switch Mechanism(An Example)", ATM Forum, ATM FORUM/95-695.

[10] ATM Forum Traffic Management Specification version 4.0, revision 10, february 1996.