

論文2000-37TC-10-3

다중 패킷 손실에서 TCP-ATM 네트워크의 성능개선 방안 (Improving Performance behavior of TCP over ATM Network in multiple losses of packets)

朴佑出*, 朴相俊**, 李丙鎬***

(Woo-Chool Park, Sang-Jun Park, and Byung-Ho Rhee)

요 약

본 논문에서는 ATM망에서의 UBR 서비스를 이용한 네트워크 구성에서의 TCP 혼잡 제어 알고리즘에 따른 성능 분석을 하였다. TCP 혼잡 제어 알고리즘에는 Slow Start, Congestion Avoidance, Fast Recovery, Fast Retransmit 등으로 이루어져 있다. TCP Reno, TCP Vanilla을 사용하여 ATM-UBR 서비스 망을 구축하여 성능을 비교 분석하였다. 그러나 Fast Retransmit, Recovery 알고리즘의 특성에 의하여 다중 패킷 손실이 발생할 때의 ATM-UBR 네트워크에서도 성능 저하 현상이 현저하게 발생하였다. 이러한 현상에 대한 해결책으로 본 논문은 새로운 Fast Retransmit, Recovery 알고리즘을 제시하여 성능을 개선하였다.

Abstract

In this paper, we analyze TCP congestion control algorithm over ATM-UBR network. TCP congestion control algorithm consists of slow start, congestion avoidance, fast recovery, fast retransmit. We analyze the ATM-UBR network service using the BSD 4.3 TCP Reno, Vanilla. However we found the fact that the characteristic of fast retransmit, recovery algorithm makes a serious degradation of performance in multiple losses of packets. We propose new fast retransmit, recovery algorithm to improve the problem. The results of performance analysis improve the multiple losses of packets using a proposed fast retransmit, recovery algorithm.

I. 서 론

Asynchronous Transfer Mode(ATM)은 고속의 멀티

* 正會員, 漢陽大學 電子工學科

(Dept. of Electronic Engineering, Hanyang Univ.)

** 正會員, 漢陽大學 電子通信電波工學科

(Division of Electrical and Computer Engineering, Hanyang Univ.)

*** 正會員, 漢陽大學 電子컴퓨터工學部

(School of Electrical Engineering and Computer Science, Hanyang Univ.)

※ 이 연구는 한양대학교 교내 연구비에 의하여 연구되었음.

接受日字:1999年12月21日, 수정완료일:2000年9月21日

미디어 네트워크 기술이다. ATM은 오디오, 비디오, 데이터 등의 멀티미디어 트래픽 전송에 매우 효율적이며 48byte의 데이터 영역과 5byte의 헤더 영역을 포함한 53byte의 고정 셀로서 오디오와 비디오 서비스에 적당한 낮은 지연 변이 특성을 가진다. 또한 ATM은 연결 지향형 서비스 특성을 가지므로 지연에 민감한 트래픽에 대하여 Quality of Service(QoS)을 보장해준다. 이러한 특징에 의하여 ATM은 고속의 멀티미디어 트래픽 전송에 가장 이상적인 기술이다.

그렇지만, ATM은 응용 서비스의 부재라는 심각한 문제점에 의하여, ATM망의 확장에 최대의 걸림돌이 되어왔다.

이러한 문제점의 해결 방안으로서 전체 LAN간 트래픽의 90% 이상을 차지하는 TCP 트래픽에 대하여

ATM망을 이용하는 연구가 많이 진행되고 있다.

ATM망에서 TCP 트래픽은 UBR, ABR 서비스의 형태로 전송되어진다.

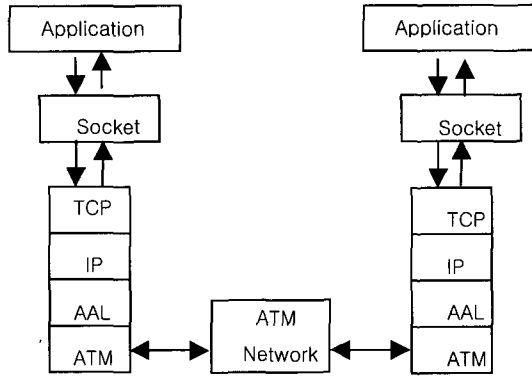


그림 1. TCP over ATM 프로토콜 스택
Fig. 1. TCP over ATM protocol stack.

초기의 연구 결과들은 주로 특정 조건하에서 TCP over ATM에서의 성능저하 현상을 나타내는 연구 결과가 주류를 이루고 있다.^[1-3]

많은 연구들을 통하여 성능 저하 현상에 따른 처리율(Throughput) 향상을 위한 여러 방안을 제시하였고, ABR, UBR 서비스의 성능 분석을 비교한 연구결과들도 많이 나와있다. 또한 LAN, WAN, 단방향, 양방향 등과 같은 많은 네트워크 환경에 따른 성능 분석도 많이 연구되었다.

본 논문에서는 ATM망에서의 UBR 서비스를 이용한 네트워크 구성에서의 TCP 혼잡 제어 알고리즘에 따른 성능 분석을 하였다.

ATM망에서의 UBR 서비스가 TCP/IP의 Best Effort 서비스 형태에 가장 적합한 형태를 가진다.^[4]

그런데 Fast Retransmit, Recovery 알고리즘의 특성에 의하여 병목 현상이 심하게 일어나는 상황에서 다중 패킷 손실이 발생할 때의 성능 저하 현상이 현저하게 발생하는 것을 알게 되었다.^[5]

그래서 이러한 현상을 해결하기 위한 방법으로 새로운 Fast Retransmit, Recovery 알고리즘을 제안하고자 한다. II장에서는 TCP 혼잡 제어 알고리즘에 관하여 설명하고, III장에서는 본 논문이 제안한 Fast Retransmit, Recovery 알고리즘에 관하여 설명한다. IV장에서는 시뮬레이션모델에 관하여 설명하고, V장에서

는 실험 결과 분석을 하였으며, VI장에서는 결론을 기술한다.

II. TCP 혼잡 제어

TCP 혼잡 제어 알고리즘에는 Slow Start, Congestion Avoidance, Fast Recovery, Fast Retransmit등으로 이루어져 있다. 현재 널리 사용되고 있는 BSD 4.3 Reno TCP 버전은 위 네 가지의 특징이 모두 구현되어 있으며, Tahoe TCP는 Fast Recovery 알고리즘이 빠져있는 것을 말한다.[6-8]

1. Slow Start & Congestion Avoidance

TCP 혼잡 제어 알고리즘에는 두 가지 상이한 모드를 가지고 있다. Slow Start 모드에서는 연결의 초기 과정에 설정되거나 패킷을 분실하여 복구 후에 CWND(congestion window)을 하나의 TCP 세그먼트의 크기로 전송하고, 각 세그먼트의 ACK 확인 후에 그 값을 두 배씩 하는 것을 말한다.

송신 측에서의 ACK을 받지 않거나, 패킷 손실이 발생했을 때는 ssthresh값을 현재 CWND의 절반 값으로 설정하고, CWND의 크기를 하나의 세그먼트 크기로 하고 Slow Start 모드를 시작한다.

Congestion 모드에서는 Slow Start 모드에서의 지수 함수적 증가와는 달리 선형적인 증가를 한다. Congestion Avoidance 모드의 시작은 CWND가 ssthresh값에 도달했을 때부터 시작이다.

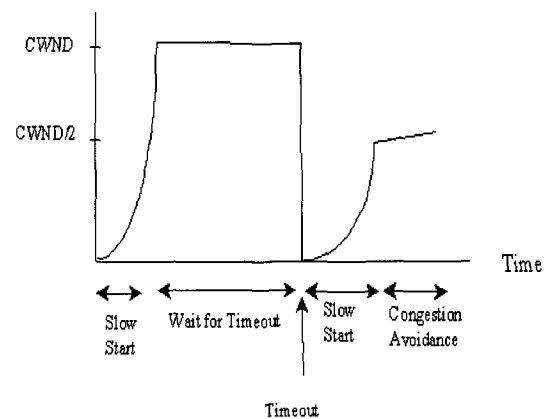


그림 2. Slow Start 와 Congestion Avoidance 알고리즘

Fig. 2. Slow Start & Congestion Avoidance Algorithm

2. Fast Retransmit & Fast Recovery

TCP 타이머는 일반적으로 500ms 알갱이 타이머를 사용한다. 결론적으로 세그먼트를 손실했을 때, 타임아웃을 위하여 많은 시간을 기다려야 한다. 이러한 기다리는 시간에 TCP 네트워크는 새로운 패킷을 전송하거나, 잃어버린 패킷을 재전송하지 못한다. 게다가 타임아웃이 발생하면 CWND는 하나의 세그먼트 크기로 설정된다.

이것의 의미는 네트워크 링크 용량을 효율적으로 사용하기 위해서는 많은 Round Trip Time을 가져야 한다는 의미이다. 이것은 낮은 효율을 의미한다. 이러한 사항에 관한 개선책으로 제시된 알고리즘이 Fast Retransmit, Fast Recovery이며, 단독 패킷 손실을 빠르게 복구하는 특성을 가진다. 수신 측에서는 순서에 맞지 않는 세그먼트를 받았을 때(손실된 세그먼트), 즉시 송신 측에 그 순서가 맞지 않은 세그먼트의 ACK을 보낸다. 송신 측에서 순서에 맞지 않은 세그먼트의 ACK를 3개 받았을 때, 그 세그먼트는 손실된 것으로 간주하고, 재전송을 한다. 이것을 fast retransmit라 한다. 송신 측에서는 ssthresh값을 현재 CWND의 절반으로 하고, 즉시 손실된 세그먼트를 재전송하고, CWND값을 $ssthresh + (3 * Maximum\ Segment\ Size)$ 로 설정한다. ACK를 받고 나서는 CWND값을 하나의 segment 만큼 증가시킨다. 이것의 의미는 송신 측에서는 네트워크 용량의 절반 정도를 유지하는 것을 말한다. 손실된 세그먼트를 재전송하고 하나의 Round Trip Time이 지난 후에, 손실된 세그먼트의 ACK를 받고 난 후에 네트워크가 적당하게 회복된 후에 CWND를 하나의 세그먼트와 Slow Start를 하지 않고, CWND를

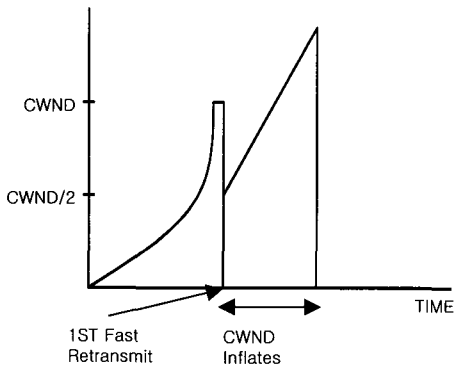


그림 3. Fast Retransmit 와 Fast Recovery 알고리즘
Fig. 3. Fast Retransmit & Fast Recovery Algorithm.

ssthresh값으로 세팅하여, Congestion Avoidance 알고리즘을 작동하는 것을 Fast Recovery라 한다.

III. 새로운 Fast Retransmit, Recovery 알고리즘

다음 그림 4는 3개 연속적인 패킷 손실 시에 congestion window의 변화를 나타낸다. 두 번의 fast retransmit 과정과 한번의 재전송 타임아웃을 가진다. 그 이유는 연속적인 패킷의 손실에 따라 두 번의 fast retransmit 이후에 특별한 duplicate Ack가 수신 측에 전달되지 않아서, 재전송 타임아웃 과정을 거치고, 이러한 과정 후, slow start 알고리즘이 작동하는데, 그림 4에서와 같이 congestion window가 1/8로 줄어든 시점에서 다시 congestion avoidance 알고리즘이 시작된다. 긴 재전송 타임아웃 기간을 거쳐야하고, 또 congestion window 크기가 많이 줄어들어 성능이 저하된다.

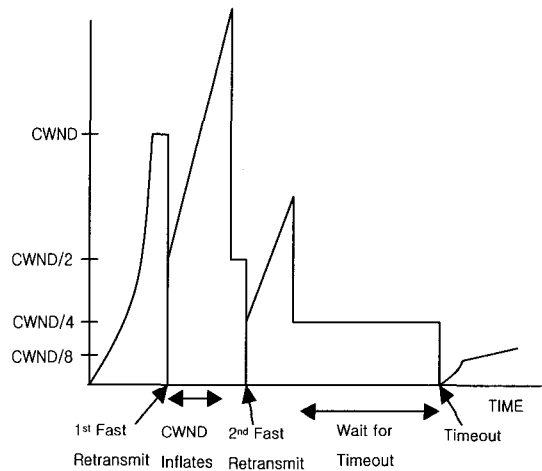


그림 4. 다중 패킷 손실시의 fast retransmit와 fast recovery 알고리즘에 의한 congestion window 변화

Fig. 4. The congestion window size of fast retransmit, recovery in multiple packet loss.

이러한 이유로 인하여, fast retransmit, recovery 알고리즘을 사용 시 다중 패킷 손실이 발생하면 효율 저하가 심각하게 된다.^[5]

다중 패킷 손실 시에는 fast retransmit로 효율적으로 회복할 수 없다는 것을 표현한다.

ATM-UBR 서비스를 이용하여 TCP/IP 서비스를 제공하면 현재 표준화 버전인 4.3 BSD Reno TCP 버전에서 다중 패킷 손실이 발생 시에 Fast Retransmit, Recovery 알고리즘에 의하여 성능이 저하되는 현상이 발견된다. 이러한 현상을 해결하기 위한 방법으로 본 논문에서는 새로운 Fast Retransmit, Recovery 알고리즘을 제안하고자 한다.

현재 TCP 혼잡 제어 알고리즘에 대해서는 많은 연구가 진행되고 있으며, 새로운 아이디어들이 계속해서 제시되고 있다.

RFC 2581, 2582 문서를^[9,10] 보면 다중 패킷 손실이 발생하였을 때 TCP 혼잡 제어 알고리즘에 관한 새로운 아이디어가 제시 되어있다. 본 논문에서는 ATM-UBR 서비스를 사용 시에 혼잡제어 알고리즘에 대한 새로운 혼잡제어 알고리즘을 제안하고자 한다.

RFC 2582에서는 세 번째 복사된 ACK를 받았을 때, 다음과 같은 식에 의하여 ssthresh가 정해진다.

$$ssthresh = \max (\text{flightSize} / 2 , 2 * \text{MSS})$$

손실된 세그먼트를 재전송하고 $cwnd = ssthresh + 3 * \text{MSS}$ 로 설정하여, 혼잡 제어 윈도우를 $3 * \text{MSS}$ 만큼 증가를 시킨다. 또한 SACK 옵션과 SACK 옵션이 없을 시에는 partial acknowledgement를 사용하였다.

이와 같이 새로운 TCP new reno를 제시하였으며, 이러한 방식은 TCP SACK option을 사용하였을 때는 다음 사항을 고려해야 한다. SACK option을 사용할 시에는 소스와 목적지 노드에서 모두 SACK 옵션을 지원해 주어야 이용이 가능하기 때문이다. NS 시뮬레이터를 이용하여^[11] 기존의 TCP reno 알고리즘과 새로운 TCP reno 알고리즘과의 성능 비교가 제시되었다. 제시된 성능 비교에 의하여 새로운 TCP reno 알고리즘이 slow-but-steady하게 대체될 것이다.

본 논문에서는 ATM 망에서 UBR 서비스를 이용한 TCP 서비스이기 때문에 ssthresh 값설정에 대하여 대역폭과 지연의 곱을 이용하여 다음과 같이 ssthresh 값을 정하게 되었다. TCP 망에서의 특성과 ATM 망에서 특성에 의하여 다중 패킷 손실이 발생했을 때를 고려하여 새로운 Fast Retransmit, Recovery 알고리즘이 필요하다. 다음은 새로운 Fast Retransmit, Recovery에 대한 설명이다.

setp 1) 과정에서 bandwidth-delay 곱을 이용하여 효

율적인 ssthresh 값을 설정하고, step 2)를 이용하여 패킷 손실 시에 slow start algorithm으로 데이터를 전송하였다. step 3)를 이용하여 재전송 타임아웃 현상을 막을 수 있다.

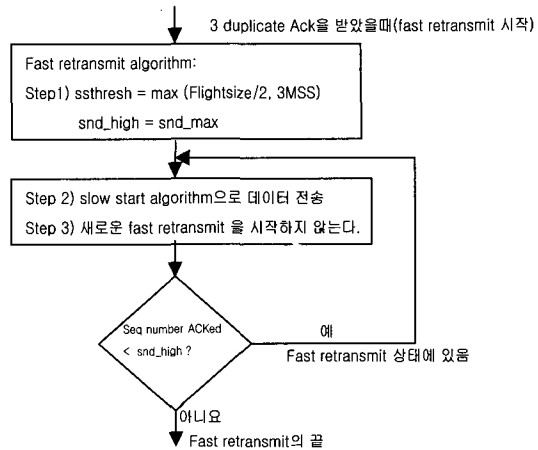


그림 5. 제안한 fast retransmit, recover 알고리즘
Fig. 5. The proposed fast retransmit, recovery algorithm.

IV. 시뮬레이션 모델 및 ATM Network

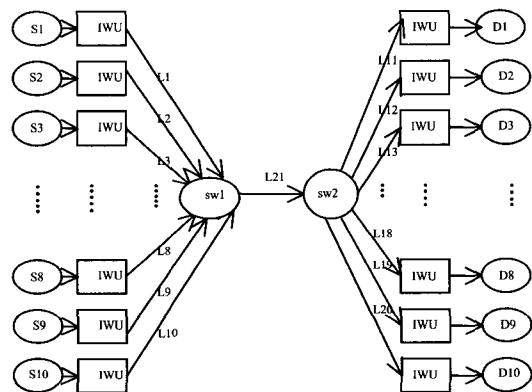


그림 6. 시뮬레이션 모델
Fig. 6. Simulation model

ATM 망에서 UBR서비스의 특징은 CBR, VBR외에 남은 용량을 사용하기 위하여 제안되었으며, 셀 손실이나 지연에 대한 민감하지 않은 서비스에 적합하다. 대역폭 부족현상이나, 사용자 특성 위한 Policy에 대하여 연결이 거절당하지 않는다. UBR 서비스^[12]에서의 스위치의 역할은 단지 큐를 모니터하고 연결 상태가 과부

하가 걸리면 셀을 버리기만 한다. 본 논문에서는 EPD 버퍼링을 사용하였으며, Nist 알고리즘 스위치^[13]을 사용하였다. 두 스위치 사이의 링크는 365566cell/sec, (155Mbps)로 하였으며, 10개의 소스가 존재하며 병목 현상이 일어나는 구조를 만들었다.

switch 1에서 큐에서 혼잡현상이 발생하여도 switch 2에는 아무런 병목현상이 발생하지 않는다.

다음 표 2는 본 논문의 시뮬레이션을 위하여 사용된 Parameter 값이다. 여기에 사용되는 Parameter 들의 값은 ATM Forum4.0^[14]를 참조하였다.

표 1. 시뮬레이션의 주요 변수
Table 1. The Parameters of Simulation.

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
I1-I20: 0.1Km, I21 : 0.3Km		

V. 결과 분석

본 논문에서의 TCP의 Slow start, Congestion Avoidance, Fast Retransmit, Fast Recovery 알고리즘을 이용하여 ATM UBR의 성능을 분석하고자 한다. 우선 Slow Start와 Congestion Avoidance 알고리즘을 사용한 TCP를 Vanilla TCP라 부르코자 한다. Slow Start, Congestion Avoidance, Fast Retransmit, Fast Recovery 알고리즘 모두를 사용한 것을 Reno TCP라 하며, 여기에서 Fast Recovery만 빠진 것을 Tahoe TCP이다. 초기 start-up 상태 동안에 혼잡제어 알고리즘에 따른 스위치1에서 셀 손실률을 표시하였다.

다음 그림은 TCP Vanilla 알고리즘을 사용하였을 때 스위치1에서의 초기 start-up 상태 동안에 셀 손실률을 나타내는 그림이다.

TCP Vanilla에서는 Slow Start, Congestion Avoidance 알고리즘을 사용한 혼잡 제어 알고리즘이다. 초기 연결이 설정될 시에 switch 1에 병목 현상에 의

하여 셀 손실률이 증가하는 현상을 알 수가 있었다. 초기 연결시 셀 손실률이 약 20% 까지 증가하는 것을 볼 수가 있다. 1초가 지난 후의 셀 손실률이 6%을 나타내는 것을 알 수가 있다.

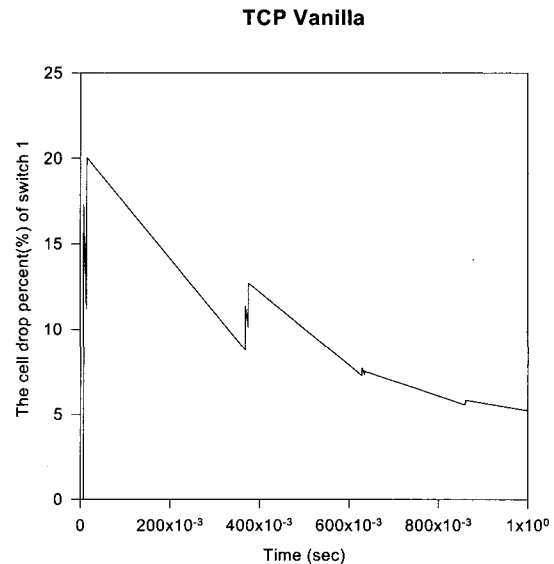


그림 7. switch 1에서의 Vanilla TCP 사용시에 초기 설정 상태에서의 셀 손실률(%).

Fig. 7. The cell drop rate(%) of switch 1 using Vanilla TCP in start-up transition.

초기의 TCP 구현 시에는 전송 중에 데이터 손실이 발생했을 때의 재전송을 위하여 Retransmit Timer Out 설정과 Go-back-n Model 만을 사용하여 혼잡제어에 사용하였다. Tahoe TCP는 이러한 초기의 TCP에 여러 가지 알고리즘들을 구현하였다. Tahoe TCP 알고리즘은 Slow Start, Congestion Avoidance, Fast Retransmit 알고리즘 등을 사용하였다.

Fast Retransmit 알고리즘은 계속해서 수많은 개선 사항들과 연구들이 진행되고 있는 알고리즘이다. Fast Retransmit 알고리즘은 송신 측에 순서에 맞지 않는 TCP 세그먼트 ACK을 연속적으로 받았을 때, Retransmit Timer Out 시간을 기다리지 않고, 그 패킷은 손실된 것으로 간주한다. 이 알고리즘의 특징은 할당된 채널을 보다 효율적으로 사용하여 처리율을 높이는 데 있다.

Reno TCP 알고리즘은 Slow Start, Congestion Avoidance, Fast Retransmit, Fast Recovery 알고리즘을 사용하였으며, 현재 BSD 4.3 Reno TCP 버전이 표

준으로 널리 사용되고 있다. Reno 버전의 특징은 Fast Retransmit에 Fast Recovery 알고리즘을 첨가하였다. 이러한 Fast Recovery 알고리즘의 특징은 Fast Retransmit 후에 발생하는 통신 채널에 효율 저하 현상을 향상시키는데 있다.

다음 그림은 Reno TCP을 사용하여 switch 1에서의 셀 손실률을 나타낸 그림이다.

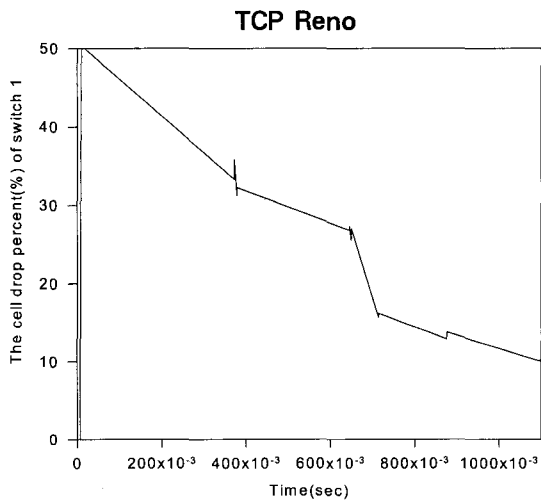


그림 8. switch 1에서의 TCP Reno 사용 시에 초기 설정 상태에서의 셀 손실율(%).

Fig. 8. The cell drop rate(%) of switch 1 using TCP Reno in start-up transition.

초기 연결 설정 시에 셀 손실률이 50%에 도달하였으며, 1초가 지난 후에도 셀 손실률이 10%가 넘는 것을 알 수가 있다.

Reno TCP 알고리즘의 특징은 단일 패킷 손실 시에 효율적인 알고리즘이나, 다중 패킷 손실 시에는 성능 저하 현상이 현저하게 일어나는 단점을 가지고 있다.

Reno TCP 알고리즘을 사용하였을 때의 현상은 Tahoe TCP 보다 switch 1에서의 셀 손실률의 저하 현상이 현저하게 나타나는 현상을 알 수가 있었다. 그 이유로는 Fast Recovery 알고리즘에서 그 원인을 분석할 수가 있다.

Fast Recovery 알고리즘의 주요한 특성은 송신 측에서 잘 못 송신된 세그먼트의 ACK을 세 개 연속해서 받았을 때, Fast Retransmit 알고리즘이 작동하며, 이것은 네트워크 연결 용량의 절반 정도를 사용하기 위하여 CWND을 ssthresh값으로 설정하여 Congestion Avoidance 알고리즘을 시작하는데 있다.

즉 다중 패킷이 손실되었을 때 CWND가 ssthresh 값으로 설정되고, 계속해서 대량의 데이터 전송에 의하여 switch 1에서의 셀 손실률이 증가하는 현상을 알 수가 있었다.

다음 그림은 Fast Retransmit, Recovery 현상에 의한 셀 손실률 증가 현상에 대한 해결 방안으로 제시한 방식의 TCP이다.

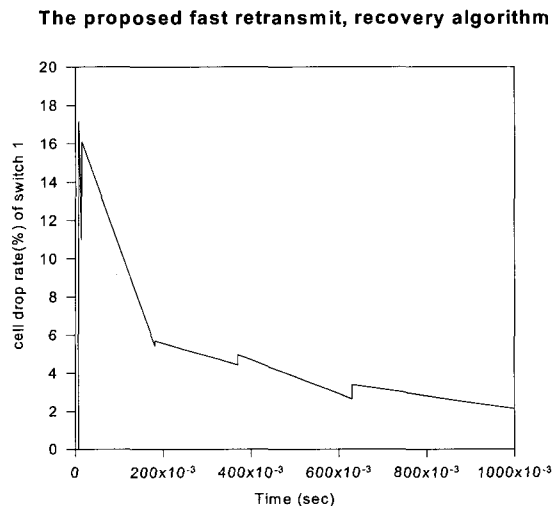


그림 9. switch 1에서의 제안한 fast retransmit, recovery TCP 사용시에 초기 설정 상태에서의 셀 손실율(%).

Fig. 9. The cell drop rate(%) of switch 1 using the proposed fast retransmit, recovery TCP in start-up transition.

ATM 망에서 UBR서비스를 사용 시에 성능 향상을 위하여 효율적인 ssthresh값을 설정하였다. 그 값은 대역폭과 지연의 곱에 의하여 구할 수 있다. 다중 패킷 손실이 발생했을 때에 fast retransmit, recovery 알고리즘에 의하여 긴 시간의 재전송 시간설정을 막았으며, ssthresh 값의 지나친 감소를 막을 수 있었다. 대역폭과 지연의 곱을 통하여 가장 효율적인 윈도우 크기 설정을 변경해 줌으로서, 초기 start-up 상태에서의 switch 1에서의 셀 손실률이 감소된 것을 알 수가 있다.

우선 초기의 셀 손실률을 보면 16% 정도이지만, 0.2 초 이후에는 5% 이하로 감소한다는 것을 알 수가 있다. 즉 셀 손실률이 Reno TCP 보다 현저하게 줄어든다는 것을 알 수가 있다.

다음 그림을 보면 TCP 혼잡 제어 알고리즘에 따라

서 수신 측에서의 평균 처리율을 알 수가 있다.

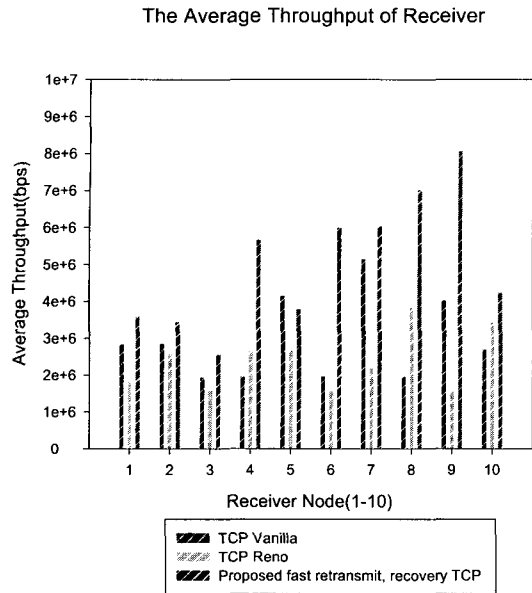


그림 10. 수신 측에서의 평균 처리율
Fig. 10. The Average Throughput of Receiver.

평균처리율 향상을 위한 효율적인 ssthresh 값을 위하여 대역폭과 지연의 곱을 계산하여 변경시켜 주었으며, 다중 패킷 손실 시에 일어날 수 있는 재전송 타임아웃을 없게 함으로써, 초기 start-up 작동상황에서 switch 1에서의 셀 손실률의 감소함에 따라서, 평균처리율이 증가한다는 현상을 알 수가 있다.

평균 처리율을 보면 Proposed fast retransmit, recovery TCP의 처리율이 가장 좋다는 것을 알 수가 있다. 본 논문이 제안한 방식을 사용함으로써, TCP Reno가 가지는 성능 저하 현상이 현저하게 향상되었다는 것을 알 수가 있다.

그 이유는 switch 1에서의 병목 현상의 발생에 의하여 다중 패킷이 손실되었을 때의 TCP Reno의 Fast Retransmit, Recovery 현상에 의하여 셀 손실이 많이 증가한다는 것을 알 수가 있다. 또한 TCP Vanilla는 네트워크의 채널 용량을 효율적으로 사용하지 못함으로써 성능이 저하되는 현상이 일어난 것을 알 수가 있다. 그래프를 살펴보면 각 수신 측의 평균처리율이 일정하지 않다는 것을 알 수가 있다. Fairness가 매우 좋지 않다는 것을 알 수가 있다. ATM-UBR의 특징인 Best Effort Service 형식이기 때문이다. 이러한 현상은 ATM-UBR의 단점이기도 하다.

VI. 결 론

본 논문에서는 ATM망에서의 TCP/IP 서비스를 할 때에 가장 TCP/IP 프로토콜에 비슷한 UBR 서비스를 이용하여 성능을 분석해 보았다. 주요한 성능의 분석요소로서 TCP에서의 혼잡 제어 알고리즘들을 분석하였다.

TCP에도 여러 가지 종류가 있으며, 본 논문에서는 TCP Vanilla, TCP Reno, 제안한 fast retransmit, recovery 알고리즘을 ATM 망을 구성해서 switch 1에서의 셀 손실률과 수신 측에서의 평균처리율을 NS 시뮬레이터를 이용하여 성능을 비교 분석하였다.

혼잡 제어 알고리즘에는 Slow Start, Congestion Avoidance, Fast Recovery, Fast Retransmit등이 있으며, 현재 널리 사용되고 있는 BSD 4.3 Reno TCP 버전인데, ATM-UBR망에서는 Fast Retransmit, Recovery 알고리즘에 의하여 다중 패킷 손실이 발생 할 때에 성능 저하 현상을 본 논문이 제안한 방식을 이용하여 송신측의 간단한 TCP 혼잡제어 알고리즘 변경에 의하여 성능을 개선 할 수가 있었다.

수신 측에서 평균처리율이 매우 일정치 않다는 현상이 발견되었는데, 이것은 ATM-UBR 서비스의 특징이라고 할 수 있다. 이러한 현상을 개선시키기 위해서는 스위치에서 각 VC 일정한 버퍼링 방법을 이용 할 수 있다.

본 논문이 제안한 방식과 스위치에서 여러 종류의 버퍼 관리 알고리즘을 사용하여 최적의 ATM-UBR 서비스의 조건에 대한 연구가 더 필요하다고 할 수 있다.

참 고 문 헌

- [1] J. S. Ahn, P. B. Danzig, Z. Liu, and L. Yan. "Evaluation of TCP Vegas: Emulation and experiment", *In Proceedings of ACM SIGCOMM '95*, pp.185~195, August 1995.
- [2] Modeklev, K. and Gunnningberg, P., "Deadlock of TCP Traffic over ATM", *In Proc. 4th Intl. IFIP workshop on Protocols for high speed networks*, pp.219~235, Vancouver, B.C., Canada, August 1994.

- [3] Allyn Romanow and Sally Floyd, "Dynamics of TCP Traffic over ATM Networks", *Computer Communication Review* Vol. 24, No. 4, pp.79~88, 1994.
- [4] 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.
- [5] Janey C. Hoe, "Improving the start-up behavior of a congestion control scheme for TCP", *In proceedings of the ACM SIGCOMM*, pp.108~120, 1996.
- [6] Fall, K and S. Floyd, "Simulation-based Comparisons of Tahoe, Reno and SACK TCP", *Computer Communication Review*, July 1996.
- [7] Stevens, W., "*TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms*", RFC 2001, January 1997.
- [8] R. Stevens, *TCP/IP Illustrated: The Protocols*, Vol. ISBN-0201-63346-9: Addison-Wesley, 1994.
- [9] M. Allman, V. Paxson, W. Stevens, "TCP Congestion Control", *RFC 2581*, April 1999.
- [10] S. Floyd and T. Henderson, "The New Reno Modification to TCP's Fast Recovery Algorithm", *RFC 2582*, April 1999.
- [11] The UCB/LBNL/VINT Network Simulator (NS). URL "http://www-mash.cs.berkeley.edu/ns/".
- [12] V. Jacobson. "Congestion avoidance and control", *In proceedings of the ACM SIGCOMM'88*, pp.314~329, August 1988.
- [13] Golmie, N., Chang, Y. and Su, D., "NIST ER Switch Mechanism(An Example)," *ATM Forum*, ATM FORUM/95-695.
- [14] The ATM Forum Traffic Management Specification version 4.0, revision 10, *ATM Forum*, February 1996.

 저 자 소 개

朴 佑 出(正會員)

1995년 2월 한양대학교 전자공학과 학사. 1997년 2월 한양대학교 전자공학과 석사. 2000년 2월 한양대학교 전자공학과 박사 과정 수료. 주관심 분야: 네트워크 프로토콜 설계 및 분석, 리눅스 임베디드 시스템. 데이터 통신. 인공 지능

朴 相 俊(正會員)

1998년 2월 한양대학교 전자공학과 학사. 2000년 2월 한양대학교 전자공학과 석사. 2000년 현재 한양대학교 전자통신전파공학과 박사과정. 주관심 분야: 네트워크 프로토콜 설계 및 분석, 리눅스 임베디드 시스템. 데이터 통신. 신호 처리. 분자 컴퓨터

李 丙 鎬(正會員) 第 36卷 S編 第 6號 參照

현재 한양대학교 전자컴퓨터공학부 교수