

TCP over ATM의 성능 개선에 관한 연구

A study on the Improvement of TCP over ATM

李璉雨*, 朴起泰*, 金珍泰**, 金來鎭***, 朴仁甲*

(Jin-Woo Lee*, Ki-Tae Park*, Jin-Tae Kim**, Nae-Jin Kim***, and In-Kap Park*)

Abstract

The Asynchronous Transfer Mode(ATM) networks are being adopted as backbones over various parts of Internet. Also, TCP is one of the most widespread transport protocols, nowadays. It can be used with ATM. But, TCP shows poor end-to-end performance on ATM networks. Effective throughput of TCP over ATM can be quite low when cells are dropped at the congested ATM switch. The low throughput is due to wasted bandwidth as congested link transmits cells from corrupted packets.

This paper examines the behavior of TCP over ATM-UBR using EPD switch in a broadband environment. With a threshold close to the buffer size, the buffer can be used more efficiently, but more drops and retransmission occurs. If the threshold is much less than buffer size, efficiency is not good, but few drops can happen. Therefore, decision of the threshold is important.

I. 서론

많은 양의 대역폭과 고속 네트워크에 대한 요구 사항이 증가함에 따라 ATM (Asynchronous Transfer Mode)이 주요한 관심사로 떠올랐다. ATM은 음성, 영상, 데이터 애플리케이션과 같은 여러 가지 형태의 트래픽을 지원하는 기능을 가지며 장거리, 근거리 망에서 사용될 수 있다. ATM은 여러 가지 형태의 서비스 클래스를 제공하는데 그 서비스들은 Constant Bit Rate(CBR), Variable Bit Rate(VBR), Available Bit Rate (ABR), 그리고 Unspecified Bit Rate(UBR)이다.

ATM에서 데이터 트래픽은 ABR과 UBR서비스에 의해 제공되며, 링크 대역폭의 할당순서는 처음에

CBR과 VBR클래스에 할당되고 남은 대역폭이 ABR과 UBR에 할당된다[1].

TCP(Transfer Control Protocol)는 데이터 전송에 사용되는 가장 일반적인 트랜스포트 프로토콜로서 두 호스트들간에 신뢰할만하며 에러가 없는 데이터의 전송을 제공한다. TCP는 흐름제어를 위해 윈도우 기반의 프로토콜을 이용하며 수신기의 윈도우(RCVWND)와 쪽주 윈도우(CWND)의 두 윈도우에 의해 시행된다. TCP는 일반적인 목적의 프로토콜이므로 LAN, MAN, WAN에서 모두 잘 동작할 수 있다[2][3].

ATM은 하드웨어와 하위 계층 소프트웨어에서는 매우 빠르게 발전해 왔지만 상위 계층 소프트웨어가 부족하다. ATM의 실제 효과를 알아보기 위해서는 ATM에서 동작하는 상위 계층 소프트웨어가 필요하지만 새로운 상위 계층 소프트웨어를 만드는데는 많은 시간이 필요하다. 따라서 현재 사용되고 있는 트랜스포트 프로토콜을 이용해 ATM망 위에서 TCP트래픽이 동작하도록 하는 것이 한가지 해결 방법이 될 수 있다.

하지만 고속 ATM망에서는 TCP가 원활하게 동작하지 않는다[4]. 이 문제는 ATM망에서 필요한 분할 재

* 建國大學校 電子工學科
(Dept. of Electronics Engineering, Konkuk Univ.)

** 네오텔레콤(Neo Telecom)

***(株) 光明制御
(KwangMyung Industrial Controls Co., Ltd.)

※ 이 논문은 1997년도 건국대학교 학술진흥연구비 지원에 의한 연구과제임

接受日: 1998年4月18日, 修正完了日: 1998年7月16日

조립에 커다란 원인이 있다. 즉, AAL(ATM Adaptation Layer)은 전송 시에 각 TCP세그먼트를 ATM셀로 분할하고 목적지에서 이 셀 들을 TCP세그먼트로 재조립한다. 이 과정에서 모든 셀들이 성공적으로 수신되었을 때만 재조립이 올바르게 처리될 수 있다. 만일 하나의 셀이 손실되면 그 세그먼트는 쓸모 없게 된다.

TCP over ATM은 ATM계층의 서비스 클래스에 따라 TCP over ATM ABR (TCP over ABR)과 TCP over ATM UBR (TCP over UBR)로 나누어진다. ABR 스위치는 좋은 성능을 보이지만 값이 비싸고 구현이 어려운 반면 UBR 스위치는 구현이 쉽지만 낮은 효율을 보인다. 따라서 UBR 스위치에 효율적인 알고리즘을 추가시켜 높은 성능을 갖도록 하는 것이 바람직할 수 있다. 본 논문에서는 EPD 방식을 사용하는 TCP over UBR의 동작 특성을 알아봄으로써 최상의 성능을 나타낼 수 있는 조건을 결정하려 하였다. 이를 위해 버퍼 요구량과 EPD 임계값의 설정에 따른 TCP 처리량의 변화를 알아보았다.

II. TCP over ATM의 동작

1. TCP 폭주제어

TCP는 두 호스트들간에 신뢰할만하며 에러가 없는 데이터의 전송을 제공하기 위해 설계된 트랜스포트 계층 프로토콜이다[2]. TCP는 애플리케이션의 데이터를 오류가 존재하는 망을 통해 전송하기 위한 수단으로써 IP를 이용하고 전송된 모든 데이터가 수신단에 오류 없이 도착하고 순서대로 재조립되도록 보장하기 위한 추가 기능들을 가지고 있다. IP와는 대조적으로 TCP는 호스트들이 데이터의 전송기간 동안 접속이 지속되도록 하고 그 후에 접속이 끊어지도록 하는 접속 지향형 프로토콜이다.

TCP는 폭주제어를 위해 윈도우 기반의 프로토콜을 이용한다. TCP접속은 망내의 패킷의 수를 제한하기 위해 종단간 흐름제어를 제공한다. 흐름제어는 두 개의 윈도우에 의해 시행되는데 수신기의 윈도우(RCVWND)는 수신기의 버퍼 용량으로써 수신기에 의

해 사용되며, 폭주 윈도우(CWND)는 망 용량의 단위로 송신측에서 유지된다.

TCP 폭주제어 방식은 "Slow Start"와 "Congestion avoidance"단계로 이루어진다[3]. 변수 Ssthresh은 두 단계간을 구별하기 위해 소스에서 유지되어진다. 소스는 한 세그먼트의 데이터를(보통 512바이트. 즉, $CWND = 1TCP\ Segment$) 전송함으로써 slow start단계에서 전송을 시작한다. 소스가 새로운 세그먼트에 대한 허가응답을 수신하면 CWND를 1만큼 증가시킨다. 세그먼트의 전송과 이에 대한 허가응답을 수신하는 시간간의 시간이 그 접속에 대한 Round Trip Time(RTT)를 지시하므로 slow start단계동안 CWND는 매 RTT마다 2배가 된다. Slow Start 단계는 CWND가 Ssthresh (보통 64K 바이트로 설정)에 이를때까지 계속되고 그 이후에 폭주 회피 단계가 시작된다. 폭주 회피 단계 동안, 소스는 세그먼트가 허가응답을 받을 때마다 CWND를 $1/CWND$ 만큼 증가시킨다. Slow start와 폭주회피 단계는 각각 매 RTT마다 지수적 증가(exponential increase)와 선형적 증가(linear increase)에 해당한다.

만일 TCP 접속이 패킷을 잃는다면, 목적지는 잘못된 순서로 수신된 패킷에 대해 연속적인 허가응답(duplicate acks)을 전송함으로써 응답한다. 소스는 마지막 허가되지 않은 패킷에 대해 재전송 타임아웃을 유지한다. 타임아웃값은 새로운 세그먼트가 허가응답되는 매시간 마다 재설정된다. 폭주는 재전송 타임아웃을 발생시킴으로써 소스에 의해 검출된다. 이 시점에서, 소스는 Ssthresh를 CWND의 절반으로 설정한다. 정확하게는 Ssthresh는 $\max\{2, \min\{CWND/2, RCVWND\}\}$ 로 설정된다.

결과적으로, $CWND < Ssthresh$ 이고 소스는 slow start 단계로 접어든다. 그러면 소스는 잃어버린 세그먼트를 재전송하고 새로운 세그먼트가 허가응답되는 매 시간마다 CWND를 1만큼 증가시킨다. 소스는 새로운 세그먼트를 전송하기 전에 잃어버린 세그먼트 이후의 세그먼트를 모두 재전송한다. 이것은 go-back-N 재전송 방식에 해당한다. 폭주 윈도우가 공시된 수신기 윈도우(RCVWND)를 넘어 증가할지라도 소스 윈도우는 둘중의 최소값에 의해 제한된다.

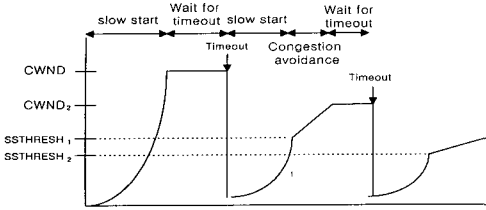


그림 1. 시간에 따른 Cwnd의 변화
Fig 1. TCP Cwnd vs Time

2. ATM 폭주제어

ATM은 동일한 네트워크상에 음성, 영상, 데이터와 같은 여러 종류의 서비스들을 제공한다. 고정된 크기의 셀과 단순한 헤더는 고속 하드웨어 스위칭을 쉽도록 한다.

ABR메카니즘은 망이 유효 트래픽 소스간에 가용 대역폭을 공정하고 효율적으로 나누도록 한다. ABR트래픽 관리에서, 소스 중단 시스템은 망이 허용하는 전송률에 의해 그들의 데이터 전송을 제한한다. 망은 소스에게 허용되는 전송률을 계산하기 위해 그들의 현재 부하 정보를 사용하는 교환기들을 이용한다. 이 전송률은 Resource Management (RM)셀을 통해 피드백으로써 소스에 전달된다.

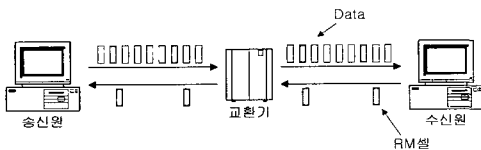


그림 2. ABR 트래픽 관리 모델
Fig 2. ABR traffic management model

ABR트래픽 관리 모델은 “rate-based end-to-end closed-loop” 모델이라고 불리운다[7]. 이 모델은 소스가 데이터를 정해진 “전송률(rate)”로 보내기 때문에 “rate-based”라고 불리운다. ABR모델은 망과 소스간에 제어정보의 연속적인 피드백을 사용하기 때문에 “close-loop”라고 불리운다. 반면에, CBR과 VBR트래픽에서는 전송률이 접속을 시작할 때 협상되고 동적으로 변화하지 않는다는 점에서 “open-loop”이다. 마지막

으로, 제어 셀이 소스로부터 목적지로 전해지고 다시 소스로 되돌려지기 때문에 “end-to-end”라고 불리운다.

순방향과 역방향에 RM셀이 계속해서 전송되면 망에는 연속적인 피드백이 존재한다. 이 상태에서, ABR 제어 루프가 설립되고 소스 전송률은 주로 망 피드백 (closed loop 제어)에 의해 제어된다. 하지만, 첫 번째 RM셀이 되돌려질 때 까지 소스의 전송률은 협상된 파라미터에 의해 제어된다.

3. TCP over ATM의 동작

그림 3은 TCP over ATM 프로토콜 구조이다. 이 프로토콜 스택은 중단간 프로토콜인 TCP에 인터페이스 되어 있는 애플리케이션을 보여준다. 데이터는 여러 계층에 의해 처리된 후에 ATM망을 통해 전송된다. 이 과정을 TCP over ATM 또는 TCP/ATM이라고 한다.

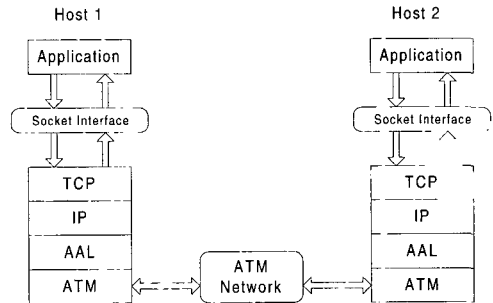


그림 3. TCP over ATM 프로토콜 스택
Fig 3. TCP over ATM Protocol Stack

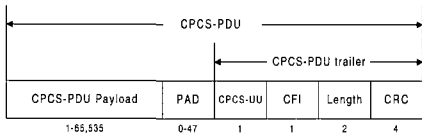
ATM계층은 AAL계층으로부터 48바이트의 유료부하를 받아 HEC를 제외한 헤더를 생성하여 유료부하에 추가한다. 또한 링크상에 들어오는 VPI(Virtual Path Identifier)와 VCI(Virtual Channel Identifier)를 적절한 출력 링크의 VCI/VPI쌍으로 해석한다.

ATM Adaptation 계층은 전송을 위해 커다란 데이터 유닛을 48바이트의 조각으로 나누는 역할을 한다. ATM상에서 전송되는 다양한 형태의 트래픽을 지원하기 위해 여러 가지 형태의 AAL이 존재하는데 TCP를 위해 가장 일반적으로 사용되는 AAL은 AAL5이다.

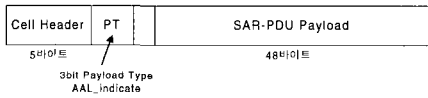
AAL5 프로토콜은 커다란 데이터 유닛(즉, TCP 세

그먼트)을 작은 셀로 나눈다. 0부터 47바이트의 padding을 데이터의 끝에 더하고 8바이트의 트레일러를 마지막 셀에 삽입한다. 트레일러는 에러검사 코드 뿐만 아니라 데이터의 전체 길이를 포함한다.

Internet Protocol(IP)은 비연결형 네트워크 계층 프로토콜이다. IP계층은 “best-effort” 를 기반으로 하여 개별 데이터그램 패킷들을 목적으로 라우팅할 책임이 있다. 종단간 에러 제어는 TCP와 같은 상위계층에 남



(1) AAL5 Common Part Convergence Sublayer (CPCS)



(2) AAL5 Segmentation and Reassembly (SAR) Sublayer

그림 4. AAL5 CPCS와 SAR포맷

Fig 4. The format of AAL5 CPCS and SAR

겨진다. 또한 IP는 전송에 적합하도록 커다란 TCP세그먼트를 작은 IP 데이터그램으로 나누는 단편화(fragmentation)을 지원하며 보통 Maximum Transmission Unit(MTU)으로 불리는 단일 유닛으로 전송될 수 있는 최대 길이의 데이터를 갖는다. AAL5에 대해 권고되는 MTU는 9180바이트이다. IP는 이보다 긴 세그먼트들을 여러개의 데이터그램으로 나눈다. 이 단편화 과정이 커다란 처리 오버헤드를 초래하므로 TCP는 모든 세그먼트의 길이가 MTU보다 작게 되도록 한다.

응용프로그램이 전송 버퍼에 데이터를 보내면 TCP는 버퍼로부터 사용자 데이터 세그먼트를 생성하고 TCP헤더를 덧붙여 이 세그먼트를 IP로 전송한다. 수신측에서는 IP로부터 데이터를 수신하고 세그먼트를 수신기 버퍼에 위치시키며, 여기에서 이들은 응용프로그램에 의해 복구될 수 있다.

응용프로그램은 망 소켓을 이용하여 TCP프로토콜과 통신을 한다. 소켓 인터페이스는 망 접속, 데이터 버퍼링, 그리고 데이터 스트림의 전송과 수신을 위한 기능을 포함한다.

4. TCP over ATM의 문제점

TCP/IP 트래픽은 일반적으로 패킷이 기본 단위인 패킷교환망을 통해 전송된다. 라우팅과 버퍼링은 패킷계층에서 수행된다. 만일 망내에서 폭주가 발생하고 교환기 버퍼가 차게 되면 전체 패킷이 버려진다. 이와는 대조적으로, ATM망은 기본 단위로 셀을 사용하고 스위칭과 버퍼링은 셀 레벨에서 발생하며 교환기 버퍼가 채워질 때 셀이 버려진다.

ATM망에서 망이 세그먼트로부터 몇 개의 셀만을 버리고 나머지 셀들을 전송하는 것은 가능하지만, 그와 같은 세그먼트에 있는 나머지 셀들은 본질적으로 “쓸모없는 것”이다. 왜냐하면 수신기가 잃어버린 셀 없이는 재구성을 할 수 없기 때문이다. “쓸모없는”셀을 전송하는 것은 이미 폭주가 발생한 망에서 대역폭을 낭비할 뿐이다. TCP소스가 잃어버린 세그먼트를 재전송하는 것은 더 많은 대역폭을 낭비하는 것이다. 결과적으로 TCP over ATM에서 낮은 처리량을 초래한다.

이에 대한 대안은 하나의 셀이 버려지자마자 ATM 교환기가 한 패킷으로부터의 모든 셀들을 버리는 것이다. 이 방법을 Partial Packet Discard(PPD)라 하며, 이것을 “selective cell discarding”이라 한다[4].

PPD는 per-VC를 기반으로 signal될 수 있다. PPD에서는 일단 교환기가 VC로부터 셀을 버리면 교환기는 AAL패킷의 끝을 지시하는 ATM셀 헤더의 ATM layer-user -to-user (AUU) 파라미터가 설정되었음을 알 때 까지는 동일한 VC로부터 계속해서 셀을 버린다. End-of-packet(EOP)셀 자체는 버려지지 않는다. AAL5는 단일 VC에서 패킷들의 동시 다중화를 지원하지 않기 때문에 AUU파라미터는 패킷 경계를 정의하는데 사용될 수 있다.

Partial Packet Discard 방식에서, 교환기는 단지 버퍼가 오버플로우될 때만 셀들을 버리기 시작하기 때문

에 개선에 한계가 있다. 교환기에 의해 버려지는 첫 번째 셀은 버퍼링된 셀이나 폭주된 링크상에 이미 전송된 셀들을 포함하는 패킷에 속할 것이다. 따라서 폭주된 링크는 파손된 패킷에 속하는 셀들의 일부분을 계속 전송할 것이다.

TCP over ATM의 유효 처리량은 버퍼가 오버플로우되기 전에 교환기가 전체 패킷을 버리면 훨씬 개선될 수 있다. Early Packet Discard 혹은 EPD라 부르는 이 방법은 폭주된 링크가 쓸모없는 셀을 전송하는 것을 방지하고 파손된 패킷의 총 수를 감소시킨다[4]. Partial Packet Discard와 마찬가지로 EPD는 per-VC를 기반으로 signal될 수 있다.

Early Packet Discard에서 교환기 버퍼가 오버플로우 되려고 하면 교환기는 AAL5프레임에 있는 모든 셀들을 버린다.

EPD방식에서, 교환기는 사용되고 있는 버퍼의 일부분이 고정된 임계값을 초과할 때마다 패킷들을 버린다. 임계값에 도달하면 교환기는 EPD를 사용하는 것으로 지정된 VC에 속하는 어떤 입력 패킷의 첫 번째 셀과 연이은 모든 셀들을 버린다. 버퍼 큐가 고정된 임계값을 초과하는 한, 교환기는 여러 접속으로부터 계속해서 패킷들을 버린다. 셀을 버릴 때 ATM교환기는 전체 셀들의 패킷을 버리는 패킷 기반의 교환기로 동작한다.

EPD에서 중요한 결정 사항은 패킷을 버리기 위한 임계값의 선택이다. 버퍼 크기와 비슷한 임계값을 설정하면 버퍼가 효율적으로 사용될 수 있지만 많은 드롭과 재전송이 발생하게 된다. 반면에, 임계값이 버퍼의 크기보다 훨씬 작으면 드롭은 적게 발생하지만 버퍼의 이용률이 낮아진다. 버퍼 임계값을 결정하는데는 두 가지 방법이 있다. 첫 번째 방법은 임계값을 전체 버퍼 크기의 일정 비율로 설정하는 것이다. 이 방법은 작은 크기의 버퍼에서는 적당하겠지만 커다란 버퍼에서는 공간을 낭비할 수도 있게 된다. 또 다른 방법은 항상 적정한 양의 공간을 예약함으로써 임계값을 설정하는 것이다. [9]에서는 임계값으로 $\{ \text{버퍼의 크기} \} - [\text{최대 세그먼트 크기(MSS)}] * [\text{활성 소스의 수}]$ 를 제시했다. 이 방식은 임계값에 도달했을 때 하나를 제외한 모든 접속이 각각의 첫 번째 셀들을 버퍼에 전송

하는 것을 최악의 경우로 가정했다. 따라서 버퍼는 $MSS * [\text{활성 소스의 수}]$ 만큼의 공간을 확보해야 할 것이다.

III. 실험 및 고찰

본 논문은 EPD 방식을 사용하는 TCP over UBR의 동작 특성을 알아보았다. 사용되는 VBR 소스의 수, EPD 임계값의 설정, 버퍼의 크기가 TCP의 성능에 어떤 영향을 미치는지 평가하였고 최상의 퍼포먼스를 갖는 조건을 결정하려 했다. 그림 5는 본 연구에서 사용된 시뮬레이션 환경이다.

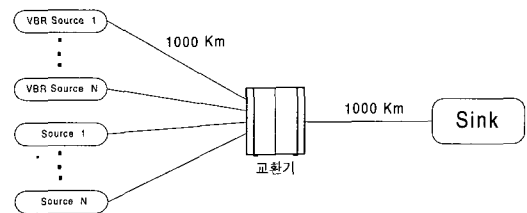


그림 5. 시뮬레이션 환경

Fig 5. Configuration of simulation

이 환경에서 backbone링크는 WAN환경의 근사화를 얻기 위해 1000Km로 구성되었으며 전체 링크는 155Mbps의 링크 속도를 갖는다. n개의 TCP 소스와 n개의 VBR 소스가 입력 송신원이며 각 송신원은 무제한의 단방향 데이터를 전송한다. 즉, TCP는 윈도우가 허용하는 한 항상 전송할 패킷을 갖는다.

1. VBR 트래픽이 존재하지 않는 경우

여기에서는 EPD 임계값을 버퍼 크기의 20%로 설정한 상태에서 버퍼의 크기가 100K, 300K, 500K, 1000K인 경우 처리량의 변화를 알아보았다. 버퍼는 ATM계층에서의 버퍼이다. 버퍼의 크기가 1000K인 경우의 처리량이 가장 좋았으며 버퍼의 크기가 처리량에 비례한다는 것을 알 수 있다.

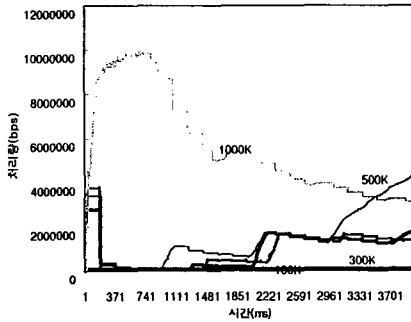


그림 6. 버퍼의 크기에 따른 처리량의 변화
 Fig 6. A performance according to the size of the buffer

이번에는 버퍼의 크기가 100K와 500K인 경우 EPD 임계값을 다르게 설정함으로써 각각에 대한 TCP 처리량의 변화를 알아보았다. EPD 임계값은 전체 버퍼 크기의 20%, 40% 그리고 [9, Goyal's UBR+]에서 제안한 방식으로 설정하였다. 버퍼의 크기가 500K인 경우 임계값을 버퍼 크기의 20%로 설정했을 때 가장 좋은 처리량을 보였으며, 100K인 경우에는 [9]에서 제안한 방식이 좋은 처리량을 보였다. 임계값을 버퍼 크기의 40%로 설정했을 때는 두 경우 모두에서 낮은 처리량을 보여 비효율적인 것으로 나타났다.

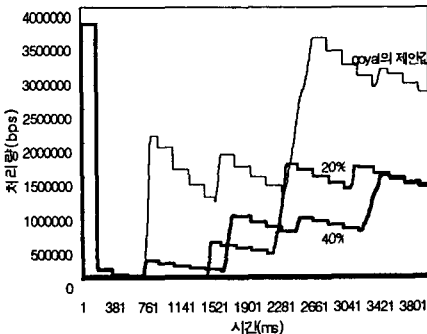


그림 7. 버퍼의 크기가 100K일 때 EPD 임계값에 따른 처리량의 변화
 Fig 7. A performance according to the EPD threshold when the size of buffer is 100K

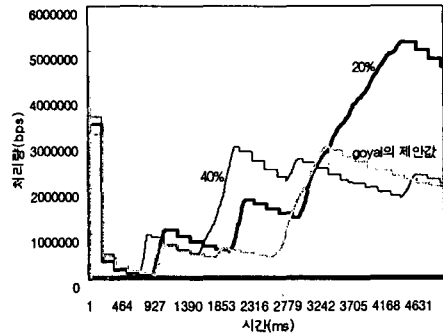


그림 8. 버퍼의 크기가 500K일 때 EPD 임계값에 따른 처리량의 변화
 Fig 8. A performance according to the EPD threshold when the size of buffer is 500K

2. VBR 트래픽이 존재하는 경우

이번에는 여러 개의 VBR 소스가 데이터를 전송하는 경우 TCP의 처리량을 알아보았다. 여러 VBR 소스가 데이터를 전송하므로 TCP 소스들은 낮은 처리량을 보일 것이다. 먼저 VBR 소스가 1인 경우와 5인 경우 버퍼의 크기에 따른 TCP 처리량을 알아보았다. 5개의 VBR 소스가 사용되었을 경우, VBR 소스들이 대부분의 대역폭을 점유함에 따라 모든 경우에 대해 TCP 소스들이 낮은 처리량을 보였다.

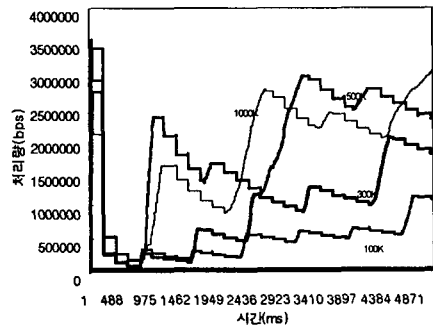


그림 9. VBR 소스가 한 개일 때 버퍼의 크기에 따른 처리량의 변화
 Fig 9. A performance according to the size of the buffer when the number of VBR source is one

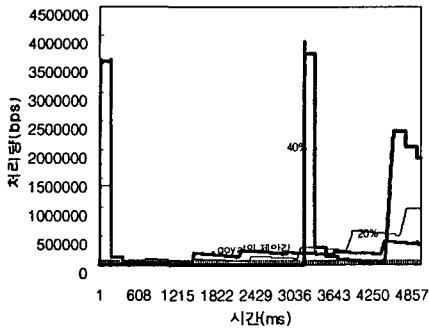


그림 10. VBR 소스의 수가 5개이고 버퍼의 크기가 100K인 경우 EPD 임계값에 따른 처리량의 변화

Fig 10. A performance according to the EPD threshold when the number of VBR source is 5 and the size of buffer is 100K

여러 개의 VBR 소스가 사용된 경우 100K와 500K의 버퍼 크기에 대하여 EPD 임계값을 다르게 설정했을 때는 임계값을 버퍼 크기의 20%로 했을 때 가장 좋은 처리량을 보였다. 5개의 VBR 소스가 사용되었을 때 버퍼 크기가 100K일 때는 임계값의 설정 방법에 관계 없이 모두 낮은 처리량을 보였으나 버퍼 크기가 500K일 때는 버퍼 크기의 20%로 임계값을 설정했을 경우 가장 좋은 처리량을 보였다.

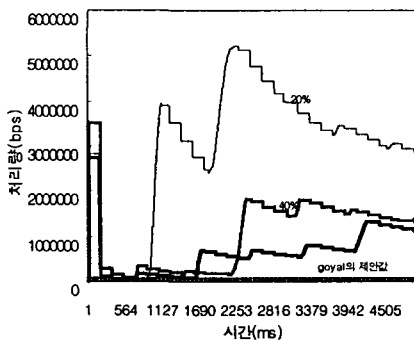


그림 11. VBR 소스의 수가 5개이고 버퍼의 크기가 500K인 경우 EPD 임계값에 따른 처리량의 변화

Fig 11. A performance according to the EPD threshold when the number of VBR source is 5 and the size of buffer is 500K

IV. 결론

TCP는 데이터 전송에 사용되는 가장 일반적인 프로토콜이며, 자체 폭주제어 메커니즘을 통해 두 호스트들간에 신뢰할만한 데이터의 전송을 제공한다. 하지만 고속 ATM망에서는 fragmentation 문제로 인해 TCP가 원활하게 동작하지 않는다.

본 논문에서는 EPD 임계값의 설정을 달리 할 경우 TCP over UBR의 특성을 알아보았다. 여러 경우에 대해서 버퍼의 크기가 TCP over ATM의 성능 향상에 중요한 요소가 된다는 것을 재확인하였고, 동일한 환경에서도 EPD 임계값을 다르게 설정함에 따라 처리량이 변화하는 것을 알 수 있었다. EPD 임계값을 버퍼의 크기에 대해 고정된 값을 사용하였을 때, 임계값이 너무 크거나 작을 경우에는 낮은 처리량을 보였다. 너무 클 경우에는 버퍼 사용 효율이 좋아지지만 셀 손실의 가능성이 존재하고 너무 낮을 경우에는 버퍼의 사용 효율이 떨어져 전체 처리량에 영향을 미치게 된다. 위 실험에서는 버퍼의 크기가 작은 경우에는 goyal의 제안값이 가장 좋은 성능을 보이지만 버퍼의 크기가 커짐에 따라 임계값이 버퍼 크기의 20% 정도가 될 때 좋은 성능을 나타냈다. 또한, 버퍼의 크기가 클수록 처리량이 향상되는 것을 알 수 있다. 즉 버퍼의 크기는 TCP over ATM의 성능 향상에 중요한 요소가 되며, 동일한 환경 내에서도 EPD 임계값을 다르게 설정함에 따라 처리량이 변화하는 것을 알 수 있다. 따라서 적절한 양의 임계값을 결정하는 것이 TCP over UBR의 성능을 결정하는데 중요한 요소가 된다.

본 논문에서는 EPD 방식을 사용하는 TCP over UBR의 성능에 영향을 미치는 EPD 임계값의 설정에 대해 알아보았다. 향후에는 각 TCP 소스들의 공정성을 보장하는 방법에 대한 연구가 진행되어야 할 것이다.

참고 문헌

- [1] "ATM Forum Traffic Management Specification Version 4.0", April 1996, <ftp://ftp.atmforum.com/pub/approved-specs/af-tm-0056.000.ps>
- [2] A. S. Tanenbaum, "Computer Networks", Prentice-Hall, 1996.

[3] W. R. Stevens, "TCP/IP Illustrated, Vol 1", Addison-Wesley, 1994.

[4] A. Romanov, S. Floyd, "Dynamics of TCP Traffic over ATM Networks", IEEE JSAC, Vol. 13, No. 4, May 1995.

[5] U. Black, "ATM: Foundation for Broadband networks", Prentice Hall, 1995.

[6] S. Kalyanaraman, R. Jain, S. Fahmy, R. Goyal, F. Lu, S. Srinidhi, "Performance of TCP/IP over ABR", Globecom '96, London, November 1996.

[7] H. Ohasaki, M. Murata, H. Suzuki, C. Ikeda, H. Miyahara, "Rate-Based Congestion Control for ATM Networks", ACM SIGCOM. 1994.

[8] C. Fang, A. Lin, "On TCP Performance of UBR with EPD and UBR-EPD with a Fair Buffer Allocation Scheme", AF-TM 95-1645, December 1995.

[9] Rohit Goyal, Raj Jain, Shiv Kalyanaraman, Sonia Fahmy, "UBR+: Improving Performance of TCP over ATM - UBR service", Proc. ICC'97, June 1997

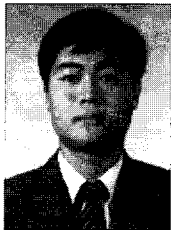
저 자 소 개



李 璣 雨 (學生會員)
 1970년 3월 28일생. 1995년 2월 경기대학교 전자공학과 졸업(공학사). 1997년 8월 건국대학교 전자공학과 대학원 졸업(공학석사). 1997년 9월 ~ 현재 건국대학교 전자공학과 박사과정 재학중. 주관심 분야는 ATM 트래픽 관리, B-ISDN 등임.



金 來 鎮 (學生會員)
 1966년 3월 21일생. 1989년 2월 건국대학교 전자공학과 졸업(공학사). 1991년 8월 건국대학교 전자공학과 대학원 졸업(공학석사). 1998년 8월 건국대학교 전자공학과 대학원 졸업예정(공학박사). 1991년 7월 ~ 현재 (주)광명제어연구소 선임연구원. 주관심분야는 DCS, SCADA등임



朴 起 泰 (學生會員)
 1969년 8월 18일생. 1993년 2월 원광대학교 전자공학과 졸업(공학사). 1996년 8월 건국대학교 전자공학과 대학원 졸업(공학석사). 1996년 9월 ~ 현재 건국대학교 전자공학과 박사과정. 1998년 4월 ~ 현재 네오텔레콤 연구원. 주관심 분야는 DSL, 무선이동통신, 무선 ATM 등임.

朴 仁 甲 (正會員)
 第 1 卷 第 1 號 參照.
 現在 建國大學校 電子工學科 教授.



金 珍 泰 (學生會員)
 1966년 3월 9일생. 1991년 2월 건국대학교 전자공학과 졸업(공학사). 1993년 8월 건국대학교 전자공학과 대학원 졸업(공학석사). 1997년 2월 건국대학교 전자공학과 대학원 박사과정 수료. 1991년 2월 ~ 1992년 3월 삼성전자 근무. 1993년 8월 ~ 1997년 8월 (주)광명제어 연구소. 1997년 9월 ~ 1998년 3월 (주)미래기술 연구소 소장. 1998년 4월 ~ 현재 네오텔레콤 책임연구원. 주관심 분야는 DSL, 무선이동통신, 실시간 분산제어 시스템, ATM등임.