ATM 전송에서 인터넷 서비스 방안

민 재 홍†

요 약

90년도 들어 WWW의 등장과 함께 TCP/IP를 이용하는 인터넷의 폭발적인 사용 증가로 인해 TCP/IP를 ATM 네트웍에서 사용하기 위한 연구가 필요하게 되었다. 현재 IP의 제한적인 기능 및 여러 가지 오버헤드로 인해 ATM 네트웍에서 IP를 효율적으로 운용하는데 많은 문제가 따르고 있다. 현재까지의 연구로는 ABR 또는 UBR 위에서 TCP/IP를 서비스 하는 방안이 제시되고 있고, ABR을 TCP/IP의 전송 수단으로 사용하는 경우 발생될 수 있는 문제점 및 스위치에서의 버퍼 요구량 등에 대한 연구가 진행 되어 왔다. 본 연구에서는 HTTP 트래픽이 갖는 특성으로 인해 TCP/ABR이 가질 수 있는 문제점을 도출하고 해결책을 제시했다. 그리고 제시된 방법들의 성능은 제한된 환경에서 시뮬레이션을 통하여 입증했다.

A Study on Guaranteeing Technique for Internet Service over ATM Transmission

Jae-Hong Min†

ABSTRACT

Recently the number of Internet users have been drastically increased with the introduction of WWW browsers, which requires accommodation study of TCP/IP on ATM networks. There are a number of problems in operating IP-over-ATM efficiently because of limited functions and overhead of IP. According to recent research, the alternatives to TCP/IP over ABR or UBR have been suggested, and the required capacity of switch buffer and problems in TCP/IP over ABR have been researched. In this paper, we address problems caused by the features of HTTP traffic and suggest a solution for TCP over ABR. The performance of the methods suggested in this paper is verified through simulation under limited conditions.

1. 서 론

1990년대 WWW의 급속한 보급과 더불어 인터넷의 이용자수가 전세계적으로 급증하고 있는 추세이며, 다양한 정보 수집과 전달 수단으로 각광 받고 있는 인터넷은 일반인들의 생활 속에 필수도구로 자리잡아 가면서, 국가의 경쟁력을 좌우할 정도로 점차 그 중요성이 커지고 있다. 이처럼 인터넷의 이용이 늘고, 인터넷을 통한 응용이 다양화되고 고도화됨에 따라 네트워크 요소들이 폭주하는 많은 양의 정보를 처리해 줄 수 있어야 할 뿐만 아니라 각각의 서비스가 요구하는 QoS(Quality of Service)를 만족시켜 줄 것을 요구한다. 그러나 기존의 장비와 TCP/IP의 제한적인 네트워크 관리 기법으로는 이러한 요구를 충족시키기가 어렵다.

그리하여 멀티미디어 정보를 통합된 구조로 전달하기 위해 B-ISDN(Broadband Integrated Service Digital Networks)개념이 등장하게 되었으며 ATM(Asynchronous Transfer Mode)을 사용하여 이를 구현하고 있다. ATM에서는 사용자 응용프로그램이 요구하는 안정적인 대역폭과 QoS를 보장해 줄 수

† 정 회 원 : 한국전자통신연구원 책임연구원
논문접수 : 1999년 9월 18일, 심사완료 : 1999년 11월 6일

있다. 그러나 이러한 이유로 기존의 모든 네트워크 기기들을 일시에 ATM구조로 교체하는 것은 비용적인 측면에서 불가능하다. 또한 새로운 장비에 대한 관리나 기존 통신 프로토콜들을 교체하는 방안들은 현존하는 TCP/IP의 보편적이고 광대한 보급으로 인해 커다란 문제점을 갖는다.

따라서 새로운 ATM 네트워크 기반 위에 기존의 TCP/IP 프로토콜을 사용하는 여러 가지 방안들이 제시되고 있다. 본 고에서는 두 개의 이질적인 계층이 연동했을 때 나타나는 현상을 ATM 서비스 클래스 중 ABR(Available Bit Rate)을 바탕으로 현재 널리 사용되는 TCP/IP 계층 위에서 동작하는 HTTP 어플리케이션을 중심으로 살펴보고 개선방안을 제시한다.

2절에서는 ATM을 하부구조로 하는 IP와의 연동 방안들을 인트러넷과 인터넷 범위에서 살펴보고, 3절에서는 ABR 트래픽 제어기법에 대해 알아본다. 4절에서는 TCP의 흐름 제어 및 혼잡(Congestion) 제어 방법에 대해 고찰하고, 5절에서는 ABR 위에 TCP를 사용했을 때 스위치 버퍼 요구량과 TCP의 처리량(throughput)을 분석해 본다. 6절에서는 HTTP 프로토콜과 Persistent TCP에 대해 알아보고, 7절에서는 앞서 언급된 각 계층이 연동될 때의 문제점과 개선방안을 논의한다.

2. IP over ATM

ATM은 그 자체로서 주소지정 방법, 계층적인 라우팅 기능, QoS 제공 등 다양한 프로토콜 특징을 가지고 있는 반면에 IP는 데이터 통신을 위한 비연결형 프로토콜임으로 다양한 어플리케이션에 대한 QoS의 제공이 미비하다. 이런 상반된 특징에도 불구하고 TCP/IP는 ATM과 공존할 수 있을 것으로 보이며, 이런 두 상반된 프로토콜의 공존을 위해서는 다음과 같은 점이 고려되어야 할 것이다:

- Packet화(Packet Encapsulation)
  IP는 자신이 수행되고 있는 아래 계층 프로토콜에 대한 아무런 정보도 가지고 있지 않다. IP 패킷은 Ethernet 위에서 수행 될 때는 Ethernet 프레임으로, Token-ring 위에서 수행될 때는 Token-ring 프레임으로 캡슐화 된다.
- 주소 변환(Address resolution)
  IP와 ATM은 서로 다른 주소 체계를 가지고 있다. Ethernet이나 Token-ring같은 공유 매체를 이용하는 LAN에서는 IP가 해당하는 MAC 주소를 얻기 위해 ARP(Address Resolution Protocol)를 사용한다. ARP는 공유매체를 이용해서 broadcast를 수행해 주소를 변환한다. 하지만 ATM은 공유 매체를 가지는 프로토콜이 아니므로 IP 주소에 해당하는 ATM 주소를 얻기 위한 방법이 제공되어야 한다.
- Broadcast와 Multicast 지원
  IP multicast는 하나 이상의 전송자가 여러 수신자에게 그룹 주소를 통해 패킷을 전송할 수 있도록 하고 있다. Broadcast는 네트워크의 모든 호스트에 패킷을 전송한다는 점에서 multicast의 변형이라고 할 수 있다. ATM은 본질적으로 broadcast와 multicast를 지원하지 않는 프로토콜이다. 이러한 사항을 고려하여 ATM 네트워크에서 IP를 효율적으로 운용하기 위한 방안으로, Intra-subnet에서 IETF의 classical IP over ATM이나 ATM Forum의 LANE 등이 소개되었고, 이를 다수의 subnet이 연결된 환경에서 운영하기 위한 방안으로 NHRP(Next Hop Resolution Protocol), MPOA(Multiprotocol Over ATM) 등이 제안되었다.
- IETF의 Classical IP over ATM
  IETF(Internet Engineering Task Force)에서 제안한 모델로 ATM을 같은 sub-net에 있는 두 호스트를 연결하는 선(wire)이나 LAN 세그먼트로 간주한다. 이 고전적인 모델은 intra-subnet의 연결만을 지원하므로 둘 이상의 IP subnet을 연결하기 위해서는 IP 라우터가 필요하다.
- ATM Forum의 LAN Emulation
  LAN Emulation은 ATM 호스트가 전통적으로 사용되어온 token-ring이나 Ethernet LAN의 프로토콜 기능을 정확히 emulation 하도록 하며, 이를 통해서 전통적인 LAN과의 internetworking이 가능하도록 한다.
- IETF의 NHRP
  NHRP는 다중 LIS(Logical Internet Subnet)환경에서 ATM 주소로부터 IP 주소를 얻어내기 위한 주소 변환 프로토콜의 일종이다. NHRP는 RFC1577에 기술된 ATMARP(ATM Address Resolution Protocol)의 확장된 버전이라고 볼 수 있으며, 이를 통해서 다른 LIS에 있는 호스트간의 ATM 직접 연결이 가

능해진다.

- ATM Forum의 MPOA
  하나의 subnet에서 여러 가지 프로토콜을 제공하는 LANE이 있지만 LANE은 여러 개의 subnet이 연결되는 internetworking을 지원하지 않는다. 이렇게 여러가지 종류의 프로토콜을 ATM을 기반으로 하는 한 개 이상의 subnet이 연결되어 있는 환경에서 사용할 수 있도록 하기 위한 ATM Forum의 작업 결과가 MPOA이다.

3. ABR 트래픽 제어

ATM Forum의 TM(Traffic Management) 4.0에서는 ATM 트래픽을 CBR(Constant Bit Rate), rt-VBR (real-time Variable Bit Rate), nrt-VBR(non-real-time Variable Bit Rate), ABR(Available Bit Rate), UBR (Unspecified Bit Rate)의 5가지 종류의 서비스로 정의하고 있다. ABR 트래픽에서는 망의 상태에 따라 송신측의 전송률을 조정할 수 있고, 가용한 네트워크 자원을 공평하고 효율적으로 사용할 수 있다. 또한, 기존 LAN에서도 작동될 수 있는 버스티(busty)하고, 독점적이며, 예측 불허한 데이터 전송을 고려하여 설계되었다. ABR은 RM(Resource Management) 셀을 통해 송신측에게 피드백을 주는 방식으로 이와 같은 형태의 트래픽을 관리한다.

ABR 트래픽 관리 모델을 전송률 기반 종단간 폐쇄루프(rate-based end-to-end closed-loop)라 부른다. 이 모델에서 전송률 기반(rate-based)이라 함은 TCP처럼 크레디트 기반(credit-based)이 아닌 송신측이 정해진 전송률로 데이터를 보낸다는 뜻이다. 또한 네트워크와 송신측간의 끊임없는 제어 정보 피드백으로 전송률을 관리하며, 송신측으로부터 수신측으로 제어 정보들을 보내고 다시 수신측으로부터 송신측으로 보내는 종단간(end-to-end) 관리기법이다.

체증 감시와 피드백 방식에 따라 여러 가지 스위치 매커니즘을 두 가지 종류로 나눌 수 있다. 하나는 EFCI (Explicit Forward Congestion Indication) 스위치이고, 다른 하나는 ER(Explicit Rate) 스위치이다. EFCI 구조에서는 체증이 발생했을 경우 해당하는 스위치에서 각 데이터 셀 헤더의 EFCI 비트를 1로 설정하고, 그럴 경우 수신측 시스템에서 역방향 RM 셀의 CI(Congestion Indication)비트를 1로 설정한다. 대부분의 경우에는 체증이 발생했는지의 여부는 큐의 길이에 따라 결정한다. 즉 큐의 길이가 한계치를 초과하였을 경우 체증이 발생했다고 판단한다. 한편 ER 피드백 구조에서는 스위치에서 VC가 지원할 수 있는 공정한 대역폭의 분배를 계산하고, 트래픽 양과 실제 Explicit rate을 결정한다. RM 셀이 지나갈 때 스위치에서 ER 필드의 결정된 Explicit rate을 설정하며 각 스위치에서는 ER 필드의 값을 증가시키지는 못한다. 이렇게 함으로써 송신측은 경로상의 모든 스위치의 허가된 MCR(Minimum Cell Rate) 값을 받게 되고 병목현상을 고려할 수 있게 된다. 이러한 ER 스위치의 구조는 EPRCA(Enhanced Proportional Rate Congestion Avoidance), ERICA(Explicit Rate Indication for Congestion Avoidance), CAPC, Charny Max-Min과 Tsang Max-Min 구조가 있다.

4. TCP 흐름/혼잡 제어

TCP에서 사용되는 흐름제어는 크레디트 할당 기법이다. 이 기법에서는 전송되는 TPDU(Transport-level protocol data unit)가 순서번호(sequence number)를 가지고 있다고 가정한다. TCP 개체(entity)는 들어오는 세그먼트를 받았음을 알리는 메시지 형태를 (A=i, W=j)와 같이 나타내고 그 의미는 다음과 같다.

- 순서번호 i-1까지의 모든 데이터를 받아 들였고, 순서번호 i의 TPDU를 다음에 받기를 원한다.
- j개의 추가적인 윈도우만큼의 데이터가 보내지도록 허가된다. 즉, 순서번호 i+j의 윈도우크기를 갖는다.

링크 제어 프로토콜로서, TCP는 흐름 제어뿐만 아니라 오류 제어 기법도 제공한다. 데이터 링크 계층 프로토콜과는 달리 TCP에서는 REJ(REJect)이나 SREJ (Selective-REJect)와 같은 부정적인 ACK는 제공하지 않는다. TCP는 긍정적인 ACK만 사용하고 주어진 시간 안에 도착하지 않았을 때만 재전송한다.

TCP의 송신 윈도우 크기는 TCP가 체증 없이 잘 작동할 수 있는지의 여부를 결정하는 중대한 문제가 된다. 현재의 TCP 구현에서 사용되고 있는 4가지 기술은 아래와 같다.

4.1 Slow start

Slow start는 혼잡 윈도우를 사용한다. 어떤 경우든지 TCP는 다음과 같은 관계를 유지하며 전송한다. 즉

크레디트 윈도우 값과 혼잡 윈도우 값 중 작은 것을 선택한다.

$$Awnd = MIN[credit, cwnd]$$

여기서,

- awnd(allowed window) : 더 이상의 ACK없이 TCP가 보낼 수 있는 TPDU의 개수를 유지하는 윈도우
- cwnd(congestion window) : TCP 연결에서 시작할 때와 혼잡으로 트래픽을 줄여갈 때 쓰이는 윈도우
- credit : 가장 최근 받아들인 ACK를 제외하고 남은 크레디트의 양

새로운 연결이 설정되었을 때, TCP 개체는 cwnd을 1로 초기화시킨다. 즉 TCP는 1개의 TPDU만을 전송하고 두 번째 TPDU를 보내기 전에 ACK를 기다려야 한다. ACK가 수신될 때마다 cwnd 값을 최대값까지 1씩 증가시킨다. 따라서 ACK 값이 지수적으로 증가함으로, 실제는 cwnd 값도 지수적으로 증가한다.

4.2 혼잡상태에서의 동적인 윈도우 크기(Dynamic window sizing on congestion)

혼잡 상태가 한번 발생하면 그러한 상태를 회복하는데 많은 시간이 걸린다. 그러므로 slow-start 기법에서 cwnd의 지수적인 증가는 너무 과다한 데이터를 전송하고 혼잡을 더욱 악화시킨다. Jacobson은 slow-start로 시작해서 다음과 같이 cwnd가 선형적으로 증가할 수 있는 방법을 제안하였다[9].

시간초과가 발생하면,

- slow-start의 한계치(threshold)를 현재 체증 윈도우의 절반으로 설정한다;

$$ssthresh = cwnd / 2$$

- cwnd = 1로 설정하고 slow-start 과정을 cwnd = ssthresh일 때까지 진행시킨다. 이 과정에서는 수신된 하나의 ACK마다 cwnd를 한 개씩 증가시킨다.
- cwnd > ssthresh일 때 각 왕복시간에 대해 cwnd를 한 개씩 증가시킨다.

4.3 빠른 재전송(Fast retransmission)

언제 TPDU를 재전송할 것인가를 결정하기 위해 재전송 타이머(RTO : retransmission timer)가 쓰인다. 이 값은 송신측에 ACK가 돌아오는 실제 왕복시간(RTT)보다 월등히 큰 값이다. 이러한 결과로 인해 만약 TPDU를 분실했다면 TCP를 재전송 하는데 시간이 많이 걸릴 것이고, 수신측 TCP가 메시지가 도착하는 순서를 지키는 정책(in-order accept policy)을 사용한다면 많은 TPDU들을 잃어버릴 것이다. Jacobson은 RTO에 의한 성능 악화를 개선시키기 위해 빠른 재전송(fast retransmit)과 빠른 회복(fast recovery)의 프로시져를 제안하였다. 빠른 재전송은 만약 TCP 수신측이 순서가 잘못된 TPDU를 받았을 때 수신측은 받은 TPDU 중 마지막으로 잘못된 것에 대해 즉시 ACK를 보낸다. TCP는 버퍼에서 빠진 TPDU를 전송 받을 때까지 마지막 TPDU에 대한 ACK를 계속해서 보낸다.

그리고 같은 TPDU에 대해 3개의 중복된 ACK(총 4개의 ACK)가 수신된 경우 시간초과에 의한 재전송을 기다리지 않고 TPDU가 분실되었다고 보고 즉시 재전송 한다.

4.4 빠른 회복(Fast recovery)

Slow-start와 혼잡 회피 프로시저는 시간초과가 발생했을 때 ssthresh를 cwnd/2로 설정하고, cwnd를 1로 설정한 후 slow-start를 시작한다. Jacobson은 이러한 과정이 지나치게 비관적인 방법이라 지적했다. 대신 잃어버린 TPDU를 재전송한 후, cwnd를 반으로 설정하고 cwnd를 선형적으로 증가시키는 빠른 회복 방안을 제시하였다. 이러한 기법은 초기의 지수적 slow-start 작업을 피하기 위해 만들어졌다.

5. TCP over ABR 모델 구성 및 평가

일반적으로 TCP와 IP는 혼잡 제어나 QoS에 대한 의존도가 높지 않다. 하지만 ATM 네트워크 위에 TCP/IP 프로토콜 스택이 점차적으로 많이 사용되면서 이러한 구조는 복잡한 QoS 기능이나 넓은 범위에서의 체증과 흐름 제어를 요구하게 되었다.

TCP/IP over ATM의 전형적인 구현은 다음과 같은 프로토콜 스택을 사용한다.

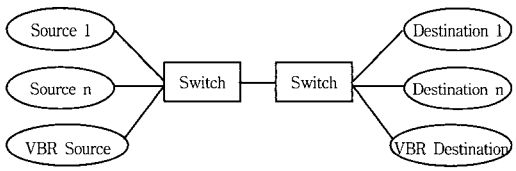
- TCP(Transmission Control Protocol)
- IP(Internet Protocol)
- AAL5(ATM Adaptation Layer 5)
- ATM

AAL은 상위층의 프로토콜 데이터를 ATM 셀로 매

칭하는 역할을 한다. AAL5는 TCP/IP 트래픽을 지원하기 위해 쓰이는 AAL 중 가장 일반적으로 쓰이는 형태이다. AAL5에서는 SAR(Segmentation And Reassembly) 단계에서 오버헤드 비트가 전혀 없다. 단순히 CS(Convergence Sublayer) 데이터를 48-옥텟의 블록으로 분할한 것으로 CS 단계의 CRC 검사를 수행하는 최소의 테일 부분만이 존재한다.

본 고에서는 TCP에서 실행되는 응용계층에서 무한한 자원 모델을 가정한다. 즉, TCP의 윈도우가 허가하는 한 TCP가 항상 보낼 패킷이 있다고 가정한다. 처리량이 경로의 길이에 제한 받지 않기 위해 윈도우의 크기를 1024KB로 정해주고 빠른 재전송과 회복은 사용하지 않는다. ABR의 송신측 파라미터는 응답과 처리량을 최대로 하기 위한 적당한 값으로 정해진다. 스위치 알고리즘은 ERICA 스위치 알고리즘을 사용하고 목표 이용률을 90%로 정한다.

그리고 전반적인 구성은 (그림 1)과 같이 "nABR 소스 + VBR 소스" 구성으로, 두 스위치간에 하나의 병목링크가 존재하며 링크용량은 nABR 소스와 VBR 소스에 의해 공유된다. 모든 링크는 155Mbps로 작동하며 길이는 1000Km이다. 여기서 VBR의 소스는 선택사항이고 대역폭에서 최대로 차지하는 비율은 80%이다. ABR은 최소한 10%의 용량을 차지하고 VBR 셀은 ABR 셀보다 높은 우선순위를 가지고 스케쥴링 된다.



(그림 1) N ABR소스 + VBR 구성도

앞에서 언급한 ABR 위의 TCP 모델을 버퍼의 설정과 ABR 용량을 변화시키며 결과를 도출하면 다음과 같다.

① 무한정 버퍼와 고정된 ABR 용량
여러 소스의 성능이 같게 나타나고 송신측은 할당된 ABR을 모두 이용할 수 있다. 결과적으로 TCP 소스들은 윈도우에 영향을 받는 것이 아니라 전송률에 제한을 받는다.

② 한정된 버퍼와 고정된 ABR 용량
소스는 전송률 제한이 아니라 윈도우에 제한을 받는다. 각각의 소스는 거의 비슷한 성능을 나타내는데 그 이유는 ABR 전송률 할당 기법이 공평하기 때문이다. ABR의 제어 루프가 설정되기 전에 처리량이 많이 떨어진다는 특징도 나타난다

③ ABR 용량에 대한 효과
VBR 트래픽을 도입하여 ABR의 용량에 영향을 미칠 수 있는 환경을 도입하고 송신측의 개수를 변화시켜 결과를 보면 다음과 같다.

- CLR(Cell Loss Rate)과 처리량 : CLR이 TCP 성능에 영향을 받지 않기 때문에 높은 셀 손실률이 TCP의 낮은 처리량과는 상관이 없다.
- 버퍼의 효과 : 무한한 TCP 소스에서는 버퍼의 크기가 클수록 더 좋은 TCP 처리량을 나타낸다. CLR에 관한 버퍼 크기의 효과는 버퍼가 클 경우 윈도우 크기가 커질 수 있고 CLR도 클 수 있다.
- 소스의 개수에 따른 영향 : 소스의 개수가 증가함에 따라 일반적으로 총 처리량도 증가한다. 이것은 더 많은 수의 소스들에 작은 크기의 윈도우를 할당하는 것이 더 많은 데이터를 전송할 수 있기 때문이다.

6. HTTP1.1의 Persistent TCP 연결

TCP의 slow start는 네트워크의 폭주를 막기 위해 고안된 매커니즘이지만, WWW 환경의 버스티한 트래픽에 적합하지 않다는 비판이 제기되고 있다. 이는 주로 한 문서의 전송 때 마다 설정되는 TCP 연결 때문이다. 즉 새로운 TCP 연결은 항상 slow start를 동반하여 이는 불필요한 지연을 초래한다. 이런 이유 때문에 persistent TCP 연결이 제안되었으며, persistent TCP 연결을 사용하는 경우 다음과 같은 장점이 있다.

- 잦은 TCP 연결의 설정과 해제를 막음으로써 CPU 시간과 TCP 프로토콜의 제어블록에 필요한 메모리를 줄일 수 있다.
- HTTP 요청과 응답을 파이프라인 형태로 사용할 수 있다. 이렇게 함으로써 하나의 TCP 연결을 좀 더 효율적으로 사용할 수 있게 된다.
- TCP 연결을 위해 사용되는 제어 패킷의 수를 줄임으로써 네트워크의 트래픽을 줄일 수 있으며, TCP에게 네트워크의 상태를 알릴 수 있는 충분한 시간을 확보해 준다.

- TCP 연결을 끊지 않은 상태에서 에러 상황을 보고할 수 있으므로, HTTP 프로토콜 자체의 개정에 도움이 된다.

HTTP 1.0과 HTTP 1.1과의 차이 중의 하나가 persistent TCP 연결이 HTTP 1.1에서는 default라는 점이다. 이를 위해 클라이언트와 서버 사이에는 TCP 연결을 끊을 수 있는 메커니즘이 존재한다. HTTP 서버는 클라이언트의 HTTP 요청에 "close token"이 포함되어 있지 않다면 일단 클라이언트가 persistent TCP 연결을 요구하고 있다고 간주한다. HTTP 클라이어트도 일단 TCP 연결이 설정되어 있을 것이라고 기대하며 만약 데이터 전송 후 연결을 끊고 싶다면 "close token"을 서버에 전송해야 한다.

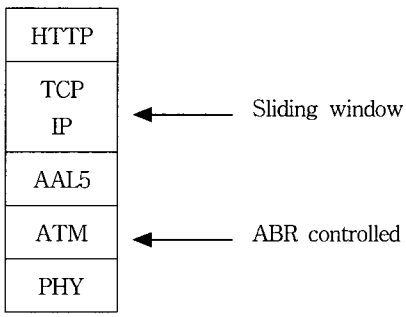
Persistent TCP 연결의 경우 클라이언트는 HTTP 요청을 파이프라이닝 할 수 있는데, 이는 여러 개의 HTTP 요청을 그에 대한 응답을 기다리지 않고 전송할 수 있음을 의미한다. Proxy 서버는 클라이언트와 서버에게 persistent 연결임을 알려야 하며, 하나의 persistent 연결은 하나의 트랜스포트 계층에만 적용되어야 한다.

서버는 일반적으로 현재 사용되지 않고 있는 연결을 끊기 위한 타이머를 가지고 있다. Proxy 서버의 경우는 클라이언트가 같은 서버로 접근할 확률이 높으므로 이 타이머 값을 매우 크게 유지한다. 하지만 클라이언트나 서버 모두 이 타이머에 대한 어떤 한계 값을 설정해 놓은 것은 아니다. 클라이언트, 서버, proxy 모두 트랜스포트 계층의 연결을 언제든지 끊을 수 있다. 이렇게 되면 클라이언트가 새로운 HTTP 요청을 요구하는 순간, 서버가 연결 해제 메시지를 보낼 수도 있게 된다. 이런 비동기적인 연결 해제에서 복구하기 위한 매커니즘을 클라이언트는 가지고 있어야 한다. Persistent TCP연결을 사용하는 클라이언트는 한 서버 당 동시에 가질 수 있는 연결의 수를 2개까지로 제한되어 있다.

7. HTTP/TCP/ABR

ATM 네트워크가 발달하면서, WWW(World Wide Web) 같은 실세계의 응용 프로그램이 ATM 네트워크에서 사용될 때의 성능에 대한 연구가 주목을 받고 있다. 이런 종류의 응용 프로그램들은 대부분 낮은 평균 대역폭만을 요구하지만 트래픽 특성이 매우 버스티하며, 이를 적절히 처리하기 위한 트래픽 관리 기법이 요구된다. WWW 응용프로그램은 데이터 전송을 위해 TCP 연결을 설정하며, TCP에게 무한대의 트래픽을 갖는 것처럼 보이는 파일 전송 프로토콜과는 달리, HTTP는 활성 주기와 비활성 주기를 갖는 버스티한 트래픽으로 보인다는 점이 매우 다르다.

앞에서 살펴본 것처럼 ABR은 데이터 전송을 위해 설계되었으며 이 절에서 제시하는 기법은 다음 (그림 2)와 같은 프로토콜 스택을 가정한다.



(그림 2) 프로토콜 스택

(그림 2)에서 볼 수 있듯이, TCP 계층의 flow/congestion 제어와 ATM 계층의 ABR close 루프 제어가 중복된다. 이 두가지 혼잡 제어 기법은 하나는 전송률에 기반하고, 다른 하나는 그레디트에 기반해서 설계되었다는 점에서 매우 다르다. 또한 두 기법의 제어 루프가 중복된다는 점도 살펴볼 만한 점 중의 하나다.

TCP를 ATM 계층 위에 사용하는 경우 고려해야 하는 혼잡에 관하여, 초기에는 UBR(Unspecified Bit Rate)을 TCP 패킷을 전달하기 위한 매커니즘으로 사용하는 것에 관한 연구가 이루어 졌는데, UBR은 트래픽을 주로 ATM 네트워크 내의 스위치에 분포하게 한다는 것이 밝혀졌다. 반면에 ABR은 ATM 네트워크의 edge device에 트래픽을 분포하게 하며, 셀 손실을 줄이기 위해서는 ATM 네트워크 내부의 스위치 보다는 ATM의 end system에 많은 버퍼를 두어야 한다는 사실이 알려졌다[1, 9].

ABR을 TCP 패킷을 전송하기 위한 메커니즘으로 사용하는데 필요한 버퍼의 양은 다양한 환경에서 시뮬레이션 되었고, 최근에는 WWW 트래픽을 이용해서 소스 트래픽을 모델링한 시뮬레이션 결과를 보이고 있다[2]. 하지만 대부분 ERICA 스위치에 대한 성능평가

에 그치고 있으며, HTTP나 TCP의 수정방안에 대한 연구는 이루어지지 않고 있다. 본 연구에서는 HTTP에서 사용되는 persistent TCP 연결의 문제점을 살펴보고 이를 해결하기 위한 방안을 제시한다.

7.1 Slow start와 Persistent TCP 연결

Slow start의 잘 알려진 단점의 하나는 global synchronization이다. 이는 네트워크가 혼잡 상태에 빠졌을 경우 모든 호스트에 ACK를 보내지 않음으로써 폭주 상태에서 벗어나려고 할 때 발생하는데, 이 경우 해당 호스트들은 동시에 slow start 상태로 빠지게 된다. 이렇게 한꺼번에 트래픽 소스들이 slow start 상태로 변하는 경우 네트워크 사용률이 매우 나빠지게 된다. 이를 해결하기 위해 RED(Random Early Detection)라는 기법이 소개되었다[9].

Slow start는 이 외에도 HTTP의 도입과 전반적인 네트워크 대역폭의 증가로 많은 비판을 받고 있다. 즉 네트워크 기술의 발달로 한 TCP 연결이 사용할 수 있는 대역폭은 넓어졌지만, slow start에서 출발하는 트래픽의 전송은 한번의 실제왕복시간(RTT) 이상의 지연을 초래하게 된다. 이런 단점을 극복하기 위해 제시된 것이 persistent TCP 연결이며, HTTP 1.0에서는 선택 사항이었지만 HTTP 1.1부터는 기본적인 요구 사항이 되었다.

ATM내의 모든 스위치는 ERICA 알고리즘을 따르고 있으며, TCP 소스와 ATM 소스는 한 호스트 내에 있다고 가정한다. 시뮬레이션 결과에 따르면 네트워크의 평균 부하가 병목 구간의 링크 용량을 초과하는 경우에는 스위치에서의 버퍼 요구량은 더 이상 증가하지 않게 된다. 이것은 ABR 피드백 제어가 매우 효율적으로 이루어지며 버스티한 WWW 트래픽에도 잘 적응한다는 것을 의미한다.

7.2 Persistent TCP 연결의 문제

ERICA 스위치 보다 빠른 응답시간을 가지지 않는 바이너리 스킴이나 EPRCA 스위치 등이 사용되는 경우 스위치에서 셀 손실이 발생할 수 있고 이는 TCP throughput에 심각한 영향을 미치게 된다. 또한 스위치 내에서의 지나친 큐 길이의 증가로 인한 RTT 증가는 QoS 보장이라는 측면에서 바람직하지 않다.

ATM 네트워크와 현재의 legacy LAN이 함께 사용되는 경우 persistent TCP 연결은 ATM edge switch에서 셀 손실을 발생시킬 위험이 있다. HTTP 연결은 persistent TCP 연결을 이용하고, 초기의 데이터 전송은 TCP의 slow start를 이용하게 된다. HTTP의 트래픽 특성이 버스티 하므로 On 구간 후에 상당히 긴 시간 동안 Off 구간이 이어진다. 만약 Off 구간이 500ms (ADTF : ACR Decrease Timeout Factor)이상 지속된다면, ABR source rule #5(Use It or Lose It : UILI)에 의해 ABR 소스는 자신의 셀 전송율을 ICR(Initial Cell Rate)으로 떨어뜨린다. 하지만 이 순간에 persistent TCP 연결은 UILI policy가 없으므로, 만약 전송 도중 패킷의 손실이 없었다면 자신의 혼잡 윈도우를 충분히 증가 시켰을 것이다. Off 구간이 끝나고 On 구간이 시작되는 순간 충분히 커져 있는 혼잡 윈도우 때문에 많은 양의 TCP 패킷이 legacy LAN을 통해서 ATM edge switch에 전달된다. 이는 ATM edge switch의 큐 길이를 증가시키며 최악의 경우에 많은 수의 클라이언트가 동시에 이런 트래픽을 ATM 네트워크에 유입 시킬 수 있고 셀 손실을 유발시킬 수 있다. ABR 트래픽 제어나 TCP 트래픽 제어 기법에서 데이터의 손실은 처리량의 급격한 감소, 지연시간 증가 등의 매우 나쁜 영향을 가져온다. ABR 소스와 TCP 소스가 한 호스트 내에 위치해 있다면, 과도하게 유입되는 TCP 패킷이 그 호스트의 하드 디스크에 버퍼링 될 수 있으므로, ATM 스위치에 주는 부담이 줄어 들 수 있다.

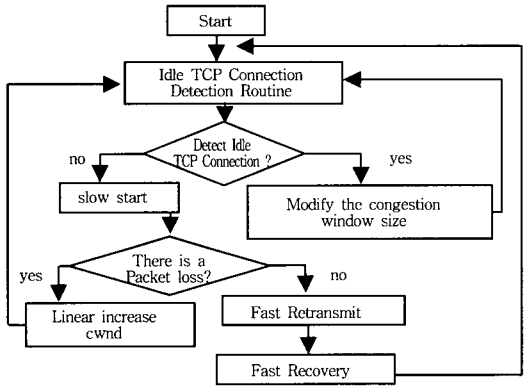
TCP 트래픽 제어 기법이 설계되었을 때는 persistent TCP 연결이나 ABR 트래픽 제어 기법 등이 등장하지 않았으므로 위와 같은 문제에 대한 고려가 이루어지지 않았다. Persistent TCP 연결은 그 자체로 네트워크의 상황을 고려하지 않고 한꺼번에 많은 트래픽을 네트워크로 전송시킬 수 있는 요소를 가지고 있다. 다음은 이에 대한 간단한 해결책을 제시한다.

7.3 새로운 TCP 혼잡 윈도우 관리 기법

ABR의 UILI policy처럼 TCP 연결도 트래픽이 일정 기간 동안 발생하지 않는다면, 자신의 혼잡 윈도우의 크기를 조절해야 한다. 이는 네트워크의 혼잡 상황을 알지 못하고 있는 상태에서 많은 트래픽이 갑자기 네트워크로 유입되는 것을 막을 수 있다. Idle 상태에 있는 persistent TCP 연결의 혼잡 윈도우의 크기를 조절하는 방법은 여러 가지가 있을 수 있다. 본 연구에서는 타이머를 사용한 2가지 수정된 TCP 혼잡 제어 기법을 제안한다.

• 지수적 윈도우 감소 기법
• TCP 계층의 UILI 기법

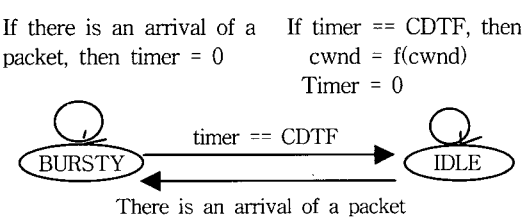
(그림 3)에 제안된 기법의 간단한 순서도를 보여주고 있다.



(그림 3) 타이머에 기반한 혼잡 윈도우 감소 기법

7.3.1 지수적 윈도우 감소 기법

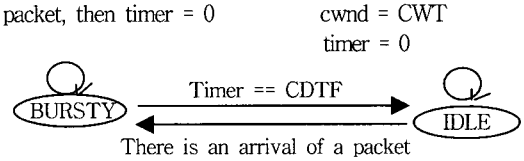
(그림 4)는 지수적 윈도우 감소 알고리즘은 사용하여 혼잡 윈도우를 줄이는 방법을 보이고 있다. 여기서 CDTF(Congestion window Decrease Timeout Factor)는 ABR의 UILI에 사용되는 ADTF(ACR Decrease Timeout Factor)에 해당하는 것으로 각 TCP 엔터티는 별도의 타이머를 이용하여 CDTF 동안 트래픽이 없다면, 혼잡 윈도우 감소 모드로 들어가게 된다. CDTF는 ADTF와 사용자의 HTML 문서 사용 패턴에 따라 설정되어야 한다[1]. 함수 f( )는 단조감소(monotonically decreasing) 함수로서 CDTF 값의 선택에 따라 여러 가지 형태를 가질 수 있다. 예를 들어 CDTF를 ADTF보다 매우 작게 선택한 경우 f( )는 선형적으로 감소하는 함수로 선택하는 것이 바람직하고, CDTF가 ADTF와 비슷한 경우는 지수적으로 감소하는 함수를 선택한다. 각각의 경우에 대해서 시뮬레이션 한 후 가장 적합한 함수를 선택해야 할 것이다.



(그림 4) 지수적 윈도우 감소기법

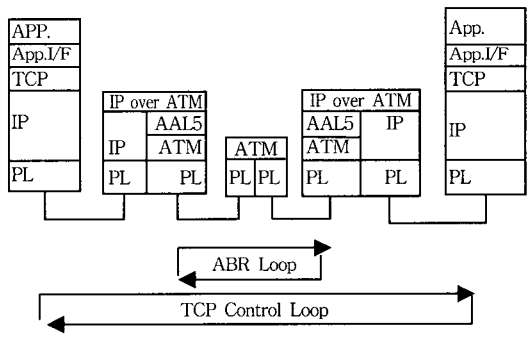
7.3.2 TCP 계층의 UILI

HTTP의 버스티한 트래픽을 ABR의 UILI 기법을 시뮬레이션 하는 방법으로 제어한다. 이 방법도 마찬가지로 타이머를 사용하지만, 혼잡 윈도우를 한번에 일정 수준(CWT : Congestion Window Threshold)으로 줄임으로써 네트워크의 과다한 패킷 유입을 막을 수 있다 이 때 CWT의 선정은 TCP에서 이미 사용되고 있는 fast recovery 방법 등을 참조하여 결정할 수 있다. (그림 5)는 이 기법의 간단한 감소 동작을 설명한다.



(그림 5) 체증 윈도우의 단순 감소 기법

7.4 시뮬레이션 분석



(그림 6) 시뮬레이션 모델

본 연구의 시뮬레이션에서는 Mah가 제시한 HTTP 네트워크 트래픽의 경험적인 모델을 사용하고, 시뮬레이션 모델은 (그림 6)과 같다[1]. HTTP 응답 길이의 누적 확률 분포는 heavy-tailed Pareto 분포로 가장 잘 나타낼 수 있으나, 확률변수 X의 최소값과 적절한 형태의 파라메터 α를 찾기가 힘들다. Pareto 분포는 다음과 같은 CDF를 갖는다.

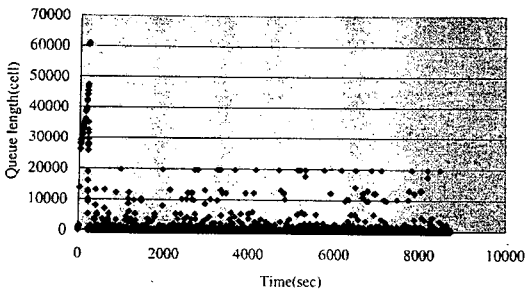
$$F(x) = P[X \leq x] = 1 - [k/x]$$

여기서 k는 X의 최소값이고, α는 형태 파라미터이다. 또한 시뮬레이션에 일반성을 부여하기 위해 HTTP

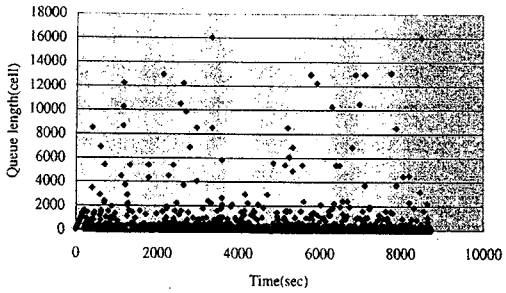
응답 길이와 사용자 think time(on/off 주기)을 데이터 파일로 사용한다. 파일 크기의 평균값은 8~10 KB, 파일크기의 중간값은 1.5~2.0 KB이다. Think time의 중간값은 약 15초이고 think time의 평균값은 약 1000초이다. ICR은 10 Mbps이고 PCR은 155.52 Mbps이며, 최대 과잉밀집 윈도우 크기는 1 Mbyte이며, 윈도우 scale 옵션을 사용한다. 또한 ADTF = 500 ms, RIF = RDF = 1/32, Nrm = 32로 가정한다. TCP 최대 세그먼트 크기는 512 byte이고 IP MTU 크기는 9180 bytes이며, 이것으로 인해 IP 세그먼트의 분할이 일어나지 않는다. 지수적으로 감소하는 알고리즘의 CWDF(Congestion Window Decreasing Factor)는 0.9이고 이 값은 CDTF = ADTF와 같다. Edge ATM 스위치에 데이터를 보내는 호스트의 개수는 일반적으로 LAN에서 사용되는 규모로 200개를 설정한다. (그림 7)에서는 ABR 위에서 동작하는 HTTP 1.1의 Persistent TCP 성능을 보여준다. (그림 7)에 나타난 것처럼 HTTP 서버에 맞물려 있는 ATM edge 스위치는 초기의 상태에서 많은 과잉밀집이 발생한다. 이 초기의 과다한 트래픽 발생은 persistent TCP의 효과와 상관없이 TCP의 과잉밀집 윈도우 메커니즘이 수정되어야 한다는 것을 암시한다. Persistent TCP 연결과 관련해서 스위치의 최대 큐 길이는 2000 셀까지 증가하는 현상이 나타나고, 최악의 경우에는 큐 길이가 순간적으로 60198 셀까지 증가한다. 본 시뮬레이션에서는 무한대의 큐 길이를 가정하고 edge 스위치의 초기 동작만을 시뮬레이션하기 때문에 셀 손실이 일어나지 않으며 중간에 위치하는 스위치들로부터 edge 스위치의 ACR를 줄이기 위한 즉각적인 피드백(negative feedback)도 없다. 그러나 불충분한 버퍼를 사용하는 경우 셀 손실이 일어난다는 것을 암시하며 이것은 TCP의 성능을 심각하게 저하시킨다.

또한 스위치가 버스트한 데이터를 수용할 수 있는 충분한 크기의 버퍼를 가지고 있더라도, 큐에서 지나치게 대기 시간이 길어질 경우 셀의 왕복지연시간(RTT)을 증가 시킨다. 버퍼의 크기가 커질수록 edge 버퍼는 대부분의 TCP 윈도우를 수용하기에 충분히 크다. 하지만 스위치의 셀 손실로 인해 EFCI를 사용하는 ABR의 처리율은 상당히 낮아질 것이다. 반면에 ER을 사용하는 ABR은 크기가 큰 버퍼를 사용하여 과잉밀집을 제어할 수 있는 edge 스위치로 과잉밀집이 분포하게 한다. 셀 손실은 edge 버퍼가 오버플로우를 발생하는 것 같은 드문 경우에만 발생한다. 패킷 손실로 인한 동적인 TCP 윈도우 관리로 인해 TCP의 흐름제어와 ATM의 흐름제어 사이의 상호 작용이 부족하게 된다. 손실된 패킷이 없을 경우에 TCP의 흐름제어에 영향을 미치는 유일한 요소는 버퍼가 채워짐에 따라 서서히 증가하며 end-to-end 지연을 증가시키는 총 왕복지연시간(RTT)이다.

수정된 TCP 과잉밀집 윈도우 관리 기법은 end-to-end 지연을 방지하기 위해 이러한 버스티한 데이터를 제한할 뿐만 아니라 SSR과 NSSR의 중간 크기로 과잉밀집 윈도우를 시작한다. (그림 8)은 지수적인 감소 기법이 큐의 길이를 17000 셀 이하로 떨어뜨림을 보여주고 있다(최악의 경우 큐 길이는 16025 셀이 된다). TCP 과잉밀집 윈도우의 지수적인 감소 기법을 사용하면 셀 손실률이 현저하게 감소하고 과잉밀집 상태도 감소하게 된다. 이런 모든 성능향상이 전체적인 TCP 처리율을 증대시킨다.



(그림 7) NSSR에서 큐 길이의 분포



(그림 8) 지수적 감소 기법에서 큐 길이의 분포

8. 결론 및 향후 연구 방향

네트워크 기술의 발달로 현재 사용되고 있는 네트워크에서 제공하지 못하는 QoS에 관한 요구사항이 늘어나고 있다. QoS를 프로토콜 자체에서 지원하는 것 중

의 하나가 ATM이며 여기서 지원하는 QoS 관련 기능들을 상위 프로토콜 계층에서 효과적으로 이용하는 것이 중요한 연구분야가 되고 있다.

현재까지는 UBR 또는 ABR을 TCP/IP의 전송 수단으로 사용하는 경우 발생할 수 있는 문제점 및 스위치에서의 버퍼 요구량 등에 대한 연구를 진행되어 왔다. 본 연구에서는 HTTP 트래픽이 갖는 특성으로 인해 TCP/ABR이 가질 수 있는 문제점을 제시하고 해결책을 제시했다. 본 과제에서 제안한 방안은 TCP 계층의 과잉 밀집 제어 윈도우를 효과적으로 제어하는 방법에 관한 것으로, 기존의 TCP 프로토콜을 많이 수정하지 않고도 구현될 수 있다는 장점이 있다. 이렇게 함으로써 기존의 TCP 프로토콜 개체와도 호환성을 가지고 동작할 수 있으며, 이는 새로운 과잉 밀집 제어 윈도우 기법이 기존의 기법을 쉽게 대체 할 수 있음을 의미한다. 새로이 제안된 윈도우 관리 기법이 효율적으로 동작할 수 있음은 7절에서 제한된 환경에서의 시뮬레이션 경과를 통해 입증했다.

본 연구에서 제시한 문제점과 그에 대한 해결책은 이런 IP스위칭을 이용한 네트워크에도 적용 가능하리라 예상되며 이에 대한 연구도 요구된다. 또한 현재 TCP 계층에서 정의된 것처럼 암시적인(implicit) 방법에 의한 과잉 밀집 상태에 대한 알림(notification)에만 의존하는 것보다는 ATM과 같은 고속 네트워크에서는 명확한(explicit) 방법에 의한 알림을 이용하는 편이 더 효율적일 수 있다. 아직 이에 대한 연구는 많이 이루어지지 않았지만 이에 대한 고려도 요구된다.

참 고 문 헌

[1] Bruce A. Mah, "An Empirical Model of HTTP Network Traffic," *INFOCOM'97*, April 1997.

[2] B. Vandalore, S. Kalyanaraman, R. Jain, R. Goyal, S. Fahmy, S. Kim, "Performance of Busty World Wide Web(WWW) Sources over ABR," Apr. 1997.

[3] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, T. Berners-Lee, "Hypertext Transfer Protocol-HTTP/1.1," REC2068, May 1997.

[4] ATM Forum, "ATM Traffic Management Specification Version 4.0," Apr. 1996.

[5] S. Kalyanaraman, R. Jain, R. Goyal, S. Fahmy, "A Survey of the Use-It-Or-Lose-It Policies for the ABR Service in Networks," submitted *to Computer Networks and ISDN Systems Journal*, 1997.

[6] Jon Postel, "Transmission Control Protocol," RFC793, Aug. 1997.

[7] S. Kalyanaraman, R. Jain, S. Fahmy, R. Goyal, S. Kim, "Performance and Buffering Requirements of Internet Protocols over ATM ABR and UBR Services," submitted to IEEE Communications Magazine, 1997.

[8] S. Kalyanaraman, R. Jain, S. Fahmy, R. Goyal, J. Jiang, S. Kim, "Performance of TCP over ABR on ATM Backbone and with various VBR Traffic Patterns," submitted to *ICC'97*, pp.8-12 Jun. 1997.

[9] S. Floyd, V. Jacobson, "Random Early Detection Gateway for Congestion Avoidance," *IEEE/ACM Trans. On Networking*, Vol.1, No.4, Aug. pp.397-413, 1993.

[10] V. N. Padmanabhan, J. C. Mogul, "Improving HTTP latency," *International WWW Conference*, Chicago, IL, Oct. 1994.

[11] S. Kalyanaraman, R. Jain, S. Fahmy, R. Goyal, S. Kim, "Buffer Requirement for TCP/IP over ABR," IEEE ATM'96, Aug. 1996.

[12] S. E. Spero, "Analysis of HTTP Performance problems," Available through, http://sunsite.unc.edu/mdmarelease/http-prob.html.

[13] Raj Jain, "Source Behavior for ATM ABR Traffic Management : An Explanation," *IEEE Communications Magazine*, Nov. 1996.

[14] Chien Fang, Arthur Lin, "Simulation Study of ABR Robustness with Binary-Mod Switches : Part2," AFTM95-1328R1, Oct. 1995.

민 재 홍

e-mail : jhmin@pec.etri.re.kr

1978년 고려대학교 산업공학과 졸업

1990년 고려대학교 경영대학원 졸업(경영정보)

1999년 한남대학교 산업정보대학원 졸업 (컴퓨터공학)

1978년~현재 한국전자통신연구원 책임연구원

관심분야 : 인터넷, 정보검색, 인공지능