

FIFO 공유 버퍼를 갖는 ATM 스위치에서 TCP 트래픽을 위한 GFR 성능 평가†

(Performance of GFR service for TCP traffic in
ATM switches with FIFO shared buffer)

박 인 용*
(Inyong Park)

요 약 ATM 포럼은 ATM 네트워크에서 TCP 트래픽에게 MCR을 보장하고, 여분의 대역폭을 공정하게 공유할 수 있도록 GFR 서비스를 정의하였다. GFR 스위치 구현 방법은 F-GCRA 알고리즘과 프레임 전달 메커니즘으로 이루어진다. F-GCRA 알고리즘은 MCR 보장에 적합한 프레임을 분류한다. 프레임 전달 메커니즘은 F-GCRA의 분류 정보에 따라 프레임 단위로 셀을 저장하고, 스케줄링 기법에 따라 출력 단자로 저장된 셀을 전달한다. 전역 임계값을 갖는 공유 버퍼로 이루어진 단순 GFR 메커니즘은 단순한 구조로 인해 구현이 용이 하지만, MCR를 보장하는데 충분치 않다고 알려져 있다. 본 논문은 단순 GFR 메커니즘을 갖는 ATM 스위치에서 TCP 트래픽을 위한 GFR 서비스 성능을 평가한다.

핵심주제어 : ATM, GFR 서비스, TCP, F-GCRA 알고리즘

Abstract ATM Forum has defined the guaranteed frame rate (GFR) service to provide minimum cell rate (MCR) guarantees for TCP traffic in ATM networks and allow it to fairly share residual bandwidth. GFR switch implementation consists of the frame-based generic cell rate algorithm (F-GCRA) and a frame forwarding mechanism. The F-GCRA identifies frames that are eligible for an MCR guarantee. The frame forwarding mechanism buffers cells at a frame unit according to information provided by the F-GCRA and forwards the buffered cells to an output port according to its scheduling discipline. A simple GFR mechanism with shared buffer with a global threshold is a feasible implementation mechanism, but has been known that it is insufficient to guarantee the MCR. This paper has estimated performance of GFR service for TCP traffic over ATM switches with the simple FIFO-based mechanism

Key Words : ATM, GFR Service, TCP, F-GCRA Algorithm

1. 서 론

Asynchronous transfer mode (ATM)은 광대역 통신을 위한 전송 기술로 개발되었지만, 최근엔

급증하는 인터넷 트래픽을 처리하기 위한 고속 연결 네트워크로 사용되고 있다. ATM은 가상 연결 (virtual connection) 사용자가 연결 설정 단계에 원하는 QoS (quality of service)를 지정하고, 필요한 네트워크 자원을 예약해야 하는 연결형 프로토콜이다. 그러나, 인터넷 트래픽의 burstness는 이러한 예약을 어렵게 한다. 그래서, 인터넷 트

† 본 연구는 금오공과대학교 학술연구비에 의하여 연구된 논문임.

* 금오공과대학교 컴퓨터공학부

래픽을 ATM 네트워크에서 효과적으로 처리하기 위하여, GFR 서비스가 ATM 포럼에서 정의되었다[1].

GFR 서비스는 가상 연결이 전송하는 프레임의 크기와 burst의 크기가 연결 설정 단계에서 협의된 maximum frame size (MFS)와 maximum burst size (MBS)를 초과하지 않는다는 가정 아래서 예약된 대역폭 MCR을 가상 연결에게 보장하는 것을 목적으로 한다. 또한, GFR 서비스는 여분의 대역폭이 있을 때, 가상 연결들이 이 대역폭을 공평하게 사용하도록 허용한다. ATM은 가변 크기의 인터넷 패킷을 고정된 크기의 ATM 셀로 나누어 전송한다. ATM 스위치가 패킷의 일부 셀을 폐기하는 경우, 나머지 셀들만으로는 패킷을 복구할 없으므로, 그 셀들은 네트워크의 대역폭을 낭비하게 된다. 이러한 단편화 (fragmentation)에 의한 대역폭 손실을 막기 위해, GFR 서비스는 ATM 계층에서 패킷의 경계를 구분할 수 있는 AAL5 PDU를 인터넷 패킷을 실어 나르는 프레임으로 사용한다. 그래서, 셀을 폐기할 경우, 그 셀이 속한 패킷의 나머지 셀들도 함께 폐기할 수 있도록 한다.

GFR 서비스를 위한 다양한 ATM 스위치 구현 방법들이 제안되었다[2,3]. 가장 간단한 구조를 갖는 단순 스위치 구현 방법은 프레임을 대역폭 보장에 적합한 것과 부적합한 것으로 나누는 F-GCRA 프레임 분류기와 가상 연결의 구분 없이 프레임 단위로 셀을 저장하는 EPD (early packet discard) 기법의 버퍼 허용 알고리즘을 FIFO 버퍼에 적용하는 것이다. 이 방법은 매우 간단하지만, 상대적으로 큰 MCR을 예약한 가상 연결에게 예약한 MCR만큼의 대역폭을 보장하지 못하는 것으로 알려져 있다[4]. 그래서, 가상 연결마다 버퍼 사용 계수기를 두고, MCR에 비례해서 버퍼를 할당하는 다양한 버퍼 허용 알고리즘들이 FIFO 버퍼를 갖는 스위치를 위해 제안 되었다 [3,5]. 또한, 가상 연결마다 별도의 버퍼를 두고 weighted fair queuing (WFQ)과 같은 전송률 보장 스케줄링을 적용하는 방법들도 제안되었다 [2,4]. 그러나, 이 구현 방법들은 GFR 서비스에 대한 만족스런 QoS를 제공할 수 있지만, 가상 연결마다 별도의 계수기와 분리된 버퍼가 필요하기 때문에, 구현 비용이 높고 운용 또한 복잡하다.

본 논문은 가상 연결 단위의 계수기나 분리된 버퍼 관리가 필요 없는 단순 FIFO 기반 GFR 스위치 구현 방법을 가지고 GFR 서비스의 QoS를 개선할 수 있는 방안을 찾기 위해, 가상 실험을 통해 단순 FIFO 기반 GFR 스위치 구현 방법의 문제점을 명확히 하고자 한다. TCP 트래픽에 대한 성능 평가를 위해, TCP 혼잡 제어 알고리즘에 따라 세 가지 TCP 구현 버전에 대해 가상 실험을 수행하였다. Tahoe 버전은 slow start와 fast retransmit 알고리즘을 채택하고 있으면, New-Reno 버전은 이 두 알고리즘 외에 개선된 fast recovery 알고리즘을 채용하고 있다. SACK 버전은 New-Reno 버전에 selective acknowledgement를 더한다. 가상 실험의 결과로 TCP 유효 처리율 (goodput), ATM 처리율 (throughput), 예약된 대역폭 사용 트래픽 분석을 통해, 단순 FIFO 기반 GFR 스위치 구현 방법의 개선점을 찾아본다.

2. 단순 GFR 스위치 구현 방법

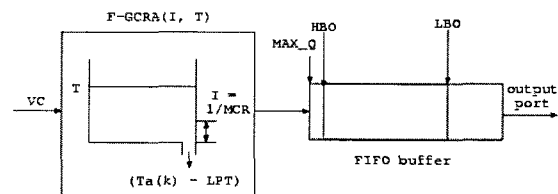


그림 1. 단순 GFR 스위치 구조

FIFO 기반 단순 GFR 스위치 구현 방법은 그림 1에서처럼 F-GCRA 프레임 분류기와 단일 FIFO 버퍼에 두 개의 광역 임계 값 (global threshold)을 갖는 버퍼 허용 알고리즘으로 이루어진다. 같은 출력 단자로 향하는 모든 가상 연결은 하나의 FIFO 버퍼를 공유한다.

2.1 F-GCRA 프레임 분류기

F-GCRA(I,T) 프레임 분류기는 가산 매개 변수 $I = 1/MCR$ 와 허용 매개 변수 $T \geq (MBS - 1) \times (1/MCR - 1/PCR) + CDVT$ 를 갖는 leaky bucket을 활용하여, 보장된 대역폭을 사용하기에 유효한 ATM 셀을 프레임 단위로 판정한다. CDVT (cell delay variation tolerance)는 허용 매

개 변수에 미치는 영향이 작아서 무시할 수 있다 [1]. 그리고, 보장된 대역폭 사용에 부적합하다고 판정된 프레임에 속하는 셀들의 cell loss priority (CLP) 비트 값을 1로 설정하는 tagging을 수행한다. 버퍼 허용 알고리즘은 셀에 대한 버퍼 허용 여부를 판단할 때, CLP 비트 값에 따라 다른 기준을 적용할 수 있다.

leaky bucket은 가상 연결마다 별도의 상태 변수를 유지한다. 상태 변수로는 bucket의 채워진 정도를 가리키는 bucket 계수기 X, 보장된 대역폭 사용에 적합한 프레임에 속하는 가장 최근에 처리한 유효 셀의 도착 시간을 가리키는 LPT, 그리고, 현재 처리하고 있는 프레임의 적합성 여부를 가리키는 tag 등이 있다.

처음에 X를 0으로 초기화하고, LPT를 첫 번째 셀의 도착 시간 $Ta(1)$ 으로 한다. $Ta(k)$ 시간에 도착한 k번째 셀이 프레임의 첫 번째 셀이라면, LPT를 $Ta(k)$ 에서 빼서, 두 셀 도착 사이의 경과 시간을 구한다. X에서 이 경과 시간을 뺀 값 X' 이 T의 값을 초과하는지 비교한다. 만약, X' 의 값이 더 크다면, F-GCRA는 그 프레임이 보장된 대역폭 사용에 부적합하다고 판정한다. 그렇지 않고, X' 의 값이 더 작다면, 그 프레임을 보장된 대역폭 사용에 적합하다고 판정한다. 프레임의 첫 번째 셀이 아닌 나머지 셀들은 첫 번째 셀의 적합성 여부를 그대로 따른다. 보장된 대역폭 사용에 적합한 프레임에 속하는 유효 셀이 도착할 때마다, X는 I만큼 증가하고, 이전 유효 셀 도착 이후 이번 유효 셀 도착 때까지 경과한 시간만큼 감소한다. 결과적으로, X는 I와 두 유효 셀들의 도착 시간 간격만큼 변한다. 그러나, X는 음의 값을 가질 수 없도록 제한된다. 보장된 대역폭 사용에 부적합한 프레임에 속하는 셀들의 CLP는 1로 tagging된다. F-GCRA 프레임 분류기의 자세한 동작은 다음과 같다.

```

초기화 X = 0, LPT = Ta(1)
가상연결 VC의 k 번째 셀 Ta(k) 시간에 도착
X' = X - (Ta(k) - LPT)
if (the first cell of a frame)
    if (X' > T)      tag = true
    else             tag = false
endif

```

```

if (Not tag)
    I = 1/MCR
    X = max(0, X') + I
    LPT = Ta(k)
else
    tag the cell

```

2.2 EPD 기법의 버퍼 허용 알고리즘

단순 GFR 스위치는 같은 출력 단자로 향하는 여러 가상 연결들이 함께 공유하는 단일 버퍼를 사용하여, 저장된 셀을 FIFO 스케줄링에 따라 출력 단자로 보낸다. 이러한 단일 FIFO 버퍼에서는 셀을 버퍼에 저장할지 혹은 폐기할지를 결정하는 버퍼 허용 알고리즘이 각 가상 연결의 링크 사용률을 조절한다. 단순 GFR 스위치는 간단한 구현을 위해 버퍼 허용 알고리즘을 적용할 때, 같은 버퍼를 공유하는 모든 가상 연결에게 공통된 광역 임계값 (global threshold)을 적용한다. 버퍼 사용량이 LBO threshold보다 작으면, 버퍼 허용 알고리즘은 도착하는 모든 셀들을 버퍼에 저장한다. 그러나, 버퍼 사용량이 LBO threshold를 넘어서면, 버퍼 허용 알고리즘은 F-GCRA 프레임 분류기에 의해 tagging이 된 셀은 더 이상 버퍼에 저장하지 않고 폐기한다. 하지만, tagging이 안된 유효 셀은 여전히 버퍼에 저장한다. 버퍼 사용량이 HBO threshold를 넘어서면, 버퍼 허용 알고리즘은 tagging이 안된 유효 셀 역시 더 이상 버퍼에 저장하지 않고 폐기한다. 그러므로, HBO threshold는 버퍼의 크기인 MAX_Q에 근접하는 값으로 설정한다.

버퍼 허용 알고리즘은 각 threshold에서 셀의 저장 여부를 결정할 때 EPD (early packet discard) 기법을 사용한다. EPD 기법은 프레임의 경계를 고려하여 프레임 단위로 셀들의 버퍼 수용 여부를 결정한다. EPD 기법은 버퍼 사용량이 threshold를 넘으면, 새롭게 도착하는 프레임에 속하는 셀을 폐기하기 시작한다. 프레임을 폐기 할 때는 그 프레임을 구성하는 첫 번째 셀부터 마지막 셀까지 모든 셀들을 폐기한다. 그러나, 버퍼 사용이 threshold를 넘기 전에 프레임의 첫 번째 셀을 이미 버퍼에 저장하였다면, 그 프레임에 속하는 나머지 셀들도 역시 버퍼에 저장한다. 또한,

EPD 기법은 이미 프레임의 첫 번째 셀을 폐기했다면, 버퍼 사용량이 threshold 보다 작아지더라도 그 프레임에 속하는 나머지 셀들 역시 모두 폐기한다. 그래서, EPD 기법은 일부 셀들을 망에서 상실하여 목적지에서 완전한 프레임을 재구성할 수 없는 불완전한 프레임에 속하는 셀들이 네트워크의 대역폭을 낭비하는 것을 막는다[6].

3. TCP 혼잡 제어 알고리즘

TCP 혼잡 제어 알고리즘의 주목적은 송신자의 전송률을 직접 제어하여 네트워크의 혼잡으로 인해 손실된 데이터 세그먼트를 재전송하는 것이다. TCP는 송신자와 수신자를 양 끝으로 하는 하나의 loop를 형성하여 이 loop에 수신자가 전송하는 수신 확인 신호(acknowledge) 정보와 윈도우, 그리고 timeout 기능을 이용하여 혼잡 제어 알고리즘을 구현하고 있다. ATM 네트워크에서 TCP 트래픽의 처리율은 네트워크가 혼잡 상황에 있을 때, TCP 송신자가 대처하는 방식에 따라 큰 차이가 있다[7]. TCP의 대표적인 혼잡 제어 알고리즘은 slow start, congestion avoidance, fast retransmit, fast recovery, selective acknowledgement 등이다.

3.1 slow start와 congestion avoidance

slow start와 congestion avoidance는 TCP 송신자가 네트워크로 내보내는 데이터 세그먼트의 양을 조절한다. 이 두 알고리즘을 구현하기 위해서는 연결 상태에 있는 TCP 세션마다 두 변수가 필요하다. 혼잡 윈도우 cwnd는 수신 확인 신호(ACK)를 받기 전까지 송신자가 보낼 수 있는 데이터 세그먼트의 양을 제한한다. 반면에 수신자 윈도우 rwnd는 수신자가 수용할 수 있는 데이터 세그먼트의 양에 대한 한계이다. 그래서, 송신자는 cwnd와 rwnd 가운데 작은 값만큼 데이터 세그먼트를 전송할 수 있다. 또 다른 상태 변수인 ssthresh (slow start threshold)는 slow start와 congestion avoidance 중에 동작할 혼잡 제어 알고리즘을 결정한다.

slow start는 처음 연결이 설정될 때 또는 혼잡에 의해 데이터 세그먼트의 손실이 발생한 경우 cwnd의 값을 1로 초기화 한다. 송신자는 수신자

로부터 ACK를 받을 때마다 cwnd의 값을 1씩 증가시킨다. 즉, 송신자가 새로운 연결을 설정하고, 하나의 데이터 세그먼트를 전송한 후 이 세그먼트에 대한 ACK를 수신하면 송신자는 두 개의 세그먼트를 전송할 수 있다. 그리고, 이 두 개의 세그먼트에 대한 ACK가 수신되면, 다시 네 개의 세그먼트를 전송할 수 있고, 다음에는 8개, 16개, 지수 적으로 증가한다. 이런 식으로 cwnd는 계속 증가하다가 임계값인 ssthresh에 이르게 되면 congestion avoidance로 동작하게 된다. ssthresh의 값은 혼잡이 발생한 때의 현재 윈도우 크기, 즉 cwnd와 rwnd 중 작은 값의 1/2로 설정한다.

congestion avoidance 상태에서는 ACK가 수신될 때마다 cwnd의 값을 $1/cwnd$ 만큼 증가시킨다. 이것은 slow start에서의 cwnd의 증가가 지수 적인데 반해서 선형적인 증가라고 할 수 있다. 즉, slow start의 경우 한번의 round trip time 동안에 수신된 ACK의 개수만큼 cwnd를 증가시키는데 반해, congestion avoidance인 경우 한번의 round trip time에 최대 한 세그먼트 크기만큼을 증가시킬 수 있는 것이다. 모든 TCP 버전은 slow start와 congestion avoidance를 포함하고 있다.

3.2 fast retransmit와 fast recovery

송신자는 TCP가 timeout 되거나 수신자가 발생시키는 중복 수신 확인 신호 DACK (duplicate ACK)을 수신했을 때, 세그먼트의 손실을 알 수 있다. 그런데, 송신자는 DACK를 수신했을 때, 과연 세그먼트가 혼잡에 의해 손실되었는지 아니면 단순히 순서가 바뀌어서 전달되었는지 알 수 없으므로 일단은 몇 개의 DACK가 도착하기까지는 기다려야 한다. 만약, 단순히 세그먼트의 순서가 뒤바뀐 것이라면 한 개에서 두 개까지의 DACK가 수신되는 동안 순서가 바뀐 세그먼트가 전달되어 새로운 ACK가 발생할 가능성이 높은 반면, retransmit threshold 이상의 DACK가 연속적으로 수신된다면 순서가 바뀐 세그먼트는 손실되었을 가능성이 높다. 그래서, retransmit threshold 이상 연속된 DACK을 수신하는 경우 TCP는 해당 세그먼트를 바로 전송하도록 하는 것을 fast retransmit라고 한다. fast retransmit은 timeout까지의 지연 시간을 줄이고, 여러 세그먼트가 재전

송되는 것을 막고, 오직 한 세그먼트만을 재전송한다.

fast recovery가 구현되기 전의 TCP에서는 DACK에 의해 fast retransmit한 이후 새로운 세그먼트를 전송하기 위해서는 다시 slow start를 실행하도록 되어 있었다. 그러나, DACK에 의해서 fast retransmit하는 경우 이미 전송 중인 세그먼트들이 존재하고 또 DACK가 발생한다는 사실은 곧 손실된 세그먼트 이외의 다른 세그먼트들은 문제없이 전송되었다는 것을 의미한다. 따라서, 새로 slow start를 통해서 설정된 연결의 안정한 상태에 도달할 필요 없이 fast retransmit한 이후 바로 congestion avoidance 상태에서 전송할 수 있도록 하는 것이 fast recovery이다.

fast retransmit은 TCP Tahoe 버전에서 구현되어, fast retransmit 후에 다시 slow start를 시작한다. fast recovery는 TCP Reno 버전에서 구현되어, fast retransmit 후에 congestion avoidance 상태에서 계속 전송한다. 하지만, 여러 개의 세그먼트가 손실된 경우, 재전송된 세그먼트에 대한 ACK는 fast retransmit가 시작되기 전에 전송된 세그먼트 중 일부만을 수신 확인할 것이다. 이러한 ACK을 PAK (partial ACK)이라고 한다. Reno 버전에서 송신자는 PAK를 받으면, fast recovery 상태에서 벗어나 cwnd의 크기를 반감한다. 손실된 세그먼트 개수만큼 반감을 하다 보면, cwnd가 작아져 재전송 timeout이 손실된 세그먼트를 복구해야 한다. 이것은 TCP의 처리율을 저하한다. 그래서, New-Reno 버전은 PAK을 받아도, fast recovery 상태를 유지하면서, PAK이 가리키는 세그먼트만을 재전송함으로써, RTT 마다 손실된 세그먼트를 하나씩 복구한다.

3.3 selective acknowledgement (SACK)

SACK 혼잡제어 알고리즘은 TCP New-Reno 버전에, 송신자가 한 윈도우 안에 여러 세그먼트를 손실했을 때, 효과적으로 복구할 수 있는 기능을 더하였다. 수신자는 수신한 세그먼트에 대한 많은 정보를 SACK에 실어 보낼 수 있다. 연속적으로 도착하는 세그먼트들 중간에 손실된 세그먼트가 있으면, SACK에 받은 세그먼트와 손실된 세그먼트를 지정하여, 송신자가 손실된 세그먼트

만을 재전송하도록 한다.

4. 실험 결과 및 성능 평가

단순 GFR 스위치 구현 방법을 적용해 만든 ATM 네트워크에서 TCP 트래픽에 대한 GFR 서비스의 QoS를 평가하기 위해 컴퓨터 모의실험을 수행하였다. 컴퓨터 모의실험 프로그램은 ATM 네트워크 연구에 널리 쓰이는 이벤트 구동 방식의 NIST ATM 시뮬레이터를 기반으로 작성하였다[8]. TCP 트래픽은 TCP Tahoe 버전, New-Reno 버전, SACK 버전에 대하여 각각 생성하였다.

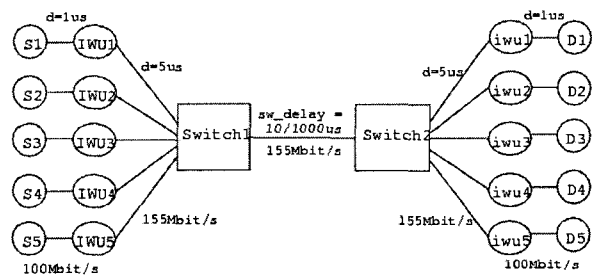


그림 2. 모의실험 환경

4.1 실험 환경

모의실험 환경은 그림 2에서 볼 수 있듯이, 5개의 송신 TCP와 5개의 수신 TCP가 100Mbit/s fast Ethernet을 통해 각각의 interworking unit (IWU)와 연결되어 있고, 각 IWU와 두 개의 ATM 스위치는 155.5Mbit/s의 OC-3c 링크로 연결되어 ATM 네트워크를 구성한다. 송신 TCP와 수신 TCP는 ATM 네트워크의 가상 연결을 통해 일대일로 연결된다. IWU는 TCP 노드로부터 IP 데이터그램으로 제조하여 TCP 노드에게 전달한다. 송신측 응용 프로그램은 각자의 상대방에게 보낼 데이터를 무한히 가지고 있는 FTP 서비스로 가정한다. MSS (maximum segment size)는 1024 바이트로 고정하고, 최대 윈도우 크기는 64K 바이트로 한다. 1024 바이트 MSS를 갖는 TCP 세그먼트에 대한 ATM 네트워크에서의 MFS (maximum frame size)는 오버헤드를 포함해서, 23 ATM 셀이다. $((1024+20(\text{TCP}))+20(\text{IP}))+8(\text{LLC}))+8(\text{AAL5}))/48=22.5$.

ATM 스위치는 같은 출력 단자를 사용하는 가상 연결들이 함께 공유하는 단일 FIFO 버퍼를 갖는다. 최대 버퍼 크기 MAX_Q는 3000 ATM 셀을 가지며, 버퍼 허용 알고리즘에서 EPD 기법을 위한 HBO threshold는 1500 ATM 셀로 한다.

4.2 성능 평가

두 ATM 스위치 사이의 전파 지연 (propagation delay) 10us와 1ms는 각각 LAN과 WAN에 대한 네트워크 환경을 고려하기 위해 적용하였다. 각 TCP 연결에 대한 TCP 유효 처리율 (goodput)으로써, TCP 계층에서 중복되지 않은 완전한 프레임 을 위해 전달된 ATM 셀의 양을 Mbit/s로 측정하였다. 또한, ATM 스위치에서 각 GFR 가상 연결에 대한 ATM 처리율을 Mbit/s로 측정하였다. 이 ATM 처리율은 F-GCRA에 의해 tagging된 프레임에 의한 여분의 대역폭 사용량과 untagged 프레임에 의한 보장된 대역폭 사용량으로 이루어진다. 모든 GFR 가상 연결의 ATM 처리량의 총합은 링크의 사용률을 나타낸다.

다섯 개의 GFR 가상 연결이 각각 5, 10, 15, 20, 25Mbit/s씩 링크에 대해 총 75Mbit/s의 MCR을 예약하는 부하가 작은 경우와 5, 10, 15, 20, 50Mbit/s씩 링크에 대해 총 100Mbit/s를 예약하는 부하가 큰 경우를 실험하였다. 모의실험으로 얻어진 결과를 각 가상 연결이 기대하는 MCR과 여분의 대역폭 (RBW)을 가상 연결로 공평하게 나누었을 때 얻을 수 있는 추가분 FS (fair share)을 더한 MCR+FS와 비교하였다.

$$RBW = Link_Capacity - \sum MCR_k$$

$$FS = 1/n \times RBW$$

각 가상 연결의 untagged 프레임이 사용한 대역폭이 MCR에 근접한다면, 예약한 MCR만큼의 최소 대역폭을 보장한다는 의미를 갖는다. 또한, ATM 처리율이 MCR+FS에 가까울수록 여분의 대역폭을 좀 더 공평하게 사용한다는 것을 의미한다.

그림 3와 그림 4는 TCP 연결이 Tahoe 버전일 경우의 각 가상 연결의 ATM 처리율 (throughput)과 TCP 유효 처리율 (goodput)을 보여준다. 먼저, untagged 프레임이 사용한 대역폭을 MCR과 비

교해 보면, 가상 연결의 MCR 값이 커질수록, 더욱 큰 격차가 벌어짐을 볼 수 있다. 이것은 단순 GFR 스위치 구현 방법이 MCR이 클수록 예약한 대역폭을 보장하는데 어려움을 갖는다는 것을 보여준다. 한편, TCP 처리율 역시 큰 MCR을 갖는 가상 연결이 MCR+FS에 못 미치는 것을 볼 수 있다. 이 또한, 예약한 대역폭만큼을 보장하지 못하고 있음을 보여준다. 특이한 사항은 ATM 처리율에 비해 TCP 유효 처리율이 큰 격차를 보이는 것이다. 이것은 TCP Tahoe 버전의 혼잡 제어 알고리즘이 매우 비효율적으로 동작한다는 것을 보여준다. ATM 처리율에 비해 TCP 유효 처리율이 낮은 까닭은 ATM 셀에 의해 전송된 많은 프레임이 중복된 프레임이기 때문이다.

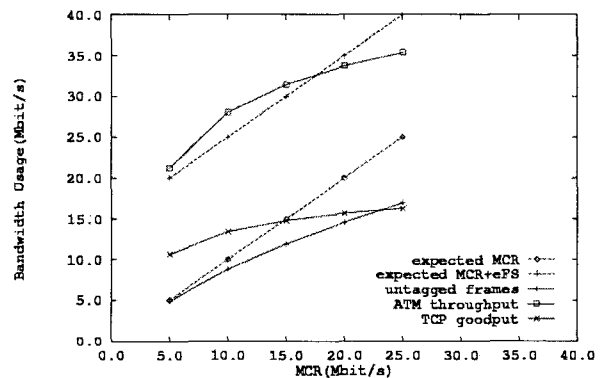


그림 3. Tahoe TCP 대역폭 사용량 (LAN)

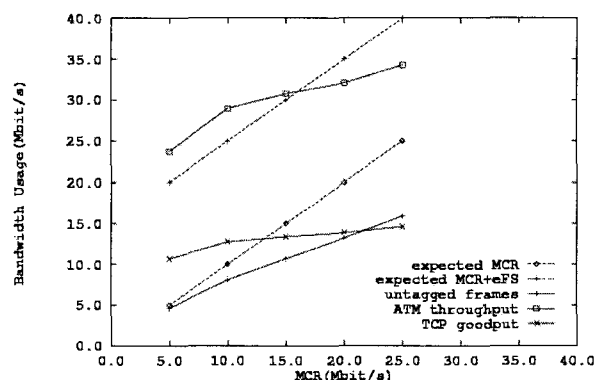


그림 4. Tahoe TCP 대역폭 사용량 (WAN)

그림 5와 그림 6은 TCP 연결이 사용하는 TCP 버전이 New-Reno인 경우의 실험 결과를 보여주고 있다. New-Reno 버전 역시 Tahoe 버전과 마

찬가지로 큰 MCR을 갖는 가상 연결일수록 untagged 프레임에 의해 사용된 대역폭이 예약한 MCR에 미치지 못함을 볼 수 있다. 한편, New-Reno에서의 TCP 유효 처리율은 ATM 처리율에 근접함을 볼 수 있다. 이것은 New-Reno의 혼잡 제어 알고리즘이 효율적으로 동작하여, 중복된 프레임의 전달을 많이 줄일 수 있었음을 보여준다. 또한, 작은 MCR을 갖는 가상 연결의 TCP 유효 처리율이 MCR+FS를 상회하는 반면, 큰 MCR을 갖는 가상 연결의 TCP 유효 처리율은 MCR+FS에 크게 못 미치는 것을 볼 수 있다. 이것은 작은 MCR을 갖는 가상 연결들이 큰 MCR을 갖는 가상 연결들 보다 여분의 대역폭을 더 많이 사용하고 있음을 의미한다. 이것으로부터 단순 GFR 스위치 구현 방법은 작은 MCR을 예약한 가상 연결이 여분의 대역폭 사용에서 더 유리하도록 허용하고 있음을 알 수 있다.

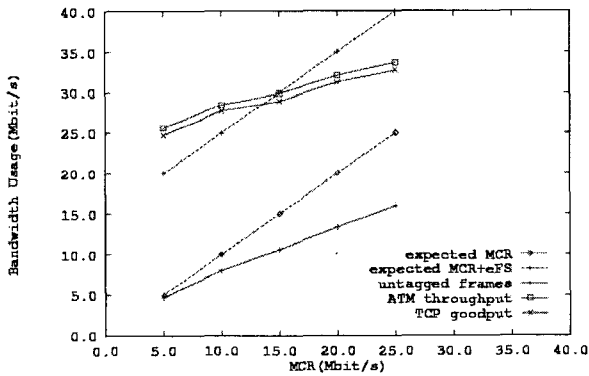


그림 5. New-Reno TCP 대역폭 사용량 (LAN)

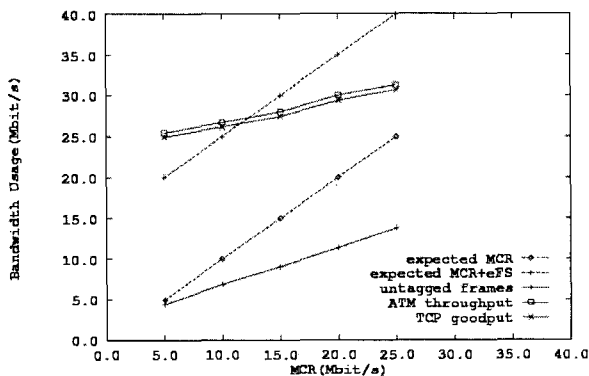


그림 6. New-Reno TCP 대역폭 사용량 (WAN)

그림 7과 그림 8은 TCP 연결이 혼잡 제어 알고리즘으로 New-Reno 버전에 SACK을 추가 구현했을 때, 얻어진 결과를 보여준다. New-Reno 버전의 모의실험과의 차이점은 TCP 유효 처리율이 ATM 처리율에 거의 근접했다는 것이다. 이것으로부터 SACK을 더한 혼잡 제어 알고리즘은 중복된 프레임이 전달되는 횟수를 최소화하고 있음을 알 수 있다.

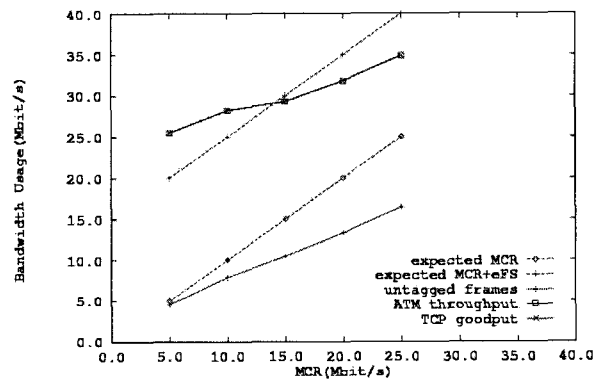


그림 7. SACK TCP 대역폭 사용량 (LAN)

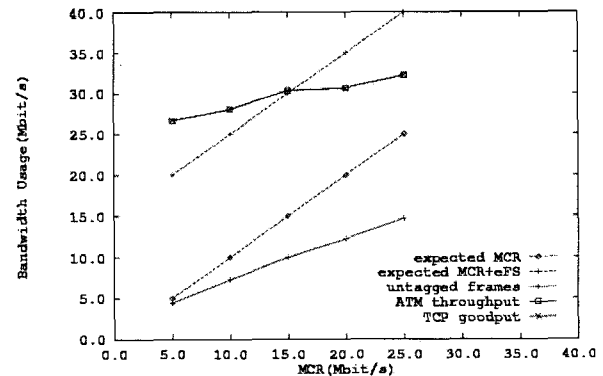


그림 8. SACK TCP 대역폭 사용량 (WAN)

그림 9와 그림 10은 혼잡 제어에 있어 가장 좋은 성능을 보이는 SACK을 사용하는 TCP 트래픽에 대한 부하가 큰 경우의 실험 결과를 보여준다. MCR로 50Mbit/s를 예약한 가상연결은 여분의 대역폭 사용량을 포함하더라도, 예약한 MCR에 크게 미치지 못하는 결과를 보여준다. 단순 GFR 스위치 구현 방법은 예약한 MCR이 클수록, QoS를 만족시키는데 어려움이 있음을 알 수 있다.

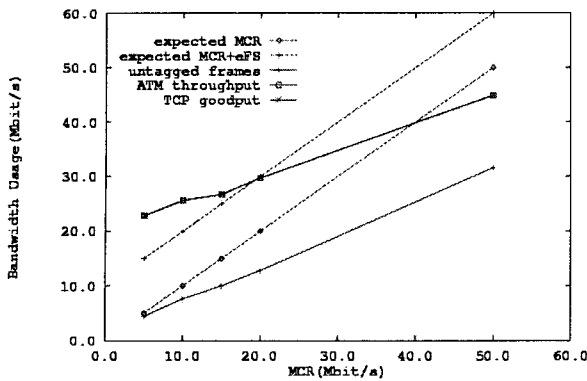


그림 9. 큰 대역폭을 예약한 경우 (LAN)

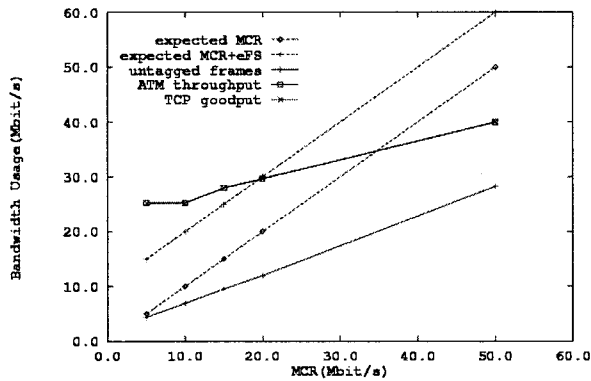


그림 10. 큰 대역폭을 예약한 경우 (WAN)

5. 결 론

본 논문은 가상 연결 단위의 계수기나 임계값 (threshold)을 필요로 하지 않는 단순 GFR 스위치 구현 방법의 각 TCP 버전에 대한 성능 평가를 컴퓨터 모의실험을 통해 수행하였다. 세 TCP 버전 중 Tahoe 버전은 중복된 프레임의 전송으로 ATM 처리율과 TCP 유효 처리율에 큰 차이를 보였다. TCP 혼잡 제어를 위해 slow start와 congestion avoidance 만으로는 충분치 않음을 확인할 수 있었다. New-Reno 버전은 ATM 처리율과 TCP 유효 처리율의 차이를 크게 줄인 개선된 결과를 보여주었다. 또한, SACK 버전의 ATM 처리율과 TCP 유효 처리율은 거의 차이가 없는 것을 보아 중복된 프레임의 전송을 최소화하여, 효율적인 혼잡 제어를 수행하고 있음을 알 수 있었다.

한편, TCP 버전에 상관없이 더 큰 MCR을 갖는 가상 연결일수록 untagged 프레임이 사용한

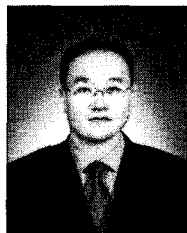
대역폭이 예약한 MCR에 더 못 미치는 것을 볼 수 있었다. 이것은 단순 FIFO 기반 GFR 스위치 구현 방법이 갖고 있는 한계를 나타낸다. 즉, 상대적으로 가장 작은 MCR을 갖는 가상 연결을 제외한 다른 가상 연결에게 예약한 MCR 만큼의 대역폭을 보장하지 못하고 있는 것이다. 이러한 현상은 F-GCRA 프레임 분류기가 필요 이상의 많은 프레임을 tagging하고 있다는 것을 보여준다. TCP 트래픽 특성에 따라 적절한 양의 프레임만을 tagging 하도록 F-GCRA 프레임 분류기가 수정될 필요가 있겠다. 또한, tagging된 프레임이 사용한 대역폭은 더 작은 MCR을 갖는 가상 연결이 더 큰 것을 확인할 수 있는데, 이 또한 EPD가 여분의 대역폭을 공평하게 할당하는데 한계가 있음을 보여준다. tagging된 프레임들에 대한 공평성을 향상시키기 위한 방안이 앞으로 연구되어야 할 것이다.

참 고 문 헌

- [1] The ATM Forum, "Traffic management specification 4.1", af-tm-00121.000, March 1999.
- [2] I. Andrikopoulos et al., "Providing rate guarantees for Internet application traffic across ATM networks", IEEE Communications Survey, 2(3), pp.2-13, 1999.
- [3] O. Bonaventure and J. Nelissen, "Guaranteed frame rate: a better service for TCP/IP in ATM networks", IEEE Networks, 15(1), pp.46-54, Jan/Feb. 2001.
- [4] Surya K. Pappu and D. Basak, "TCP over GFR implementation with different service disciplines: simulation study", ATM Forum Contribution 97-310, May 1997.
- [5] Chia-Tai Chan et al., "A FIFO-based buffer management approach for ATM GFR service", IEEE Communications Letters, 4(6), pp.205-207, 2000.
- [6] A. Romanow and S. Floyd, "Dynamic of TCD traffic over ATM networks", IEEE Journal on Selected Areas in Communications,

13(4), pp.633-641, 1995.

- [7] K. Fall and S. Floyd, "Simulation-based comparisons of Tahoe, Reno, and SACK TCP", Computer Communication Review, 26(3), pp.5-21, 1996.
- [8] Nada Golmie et al., "The NIST ATM/HFC network simulator", National Institute of Standards and Technology, 1998.



박 인 용 (Inyong Park)

- 정회원
- 1990년 2월 : 연세대학교 전산학과 (이학사)
- 1992년 2월 : 한국과학기술원 전산학과 (공학석사)
- 2002년 2월 : 한국과학기술원 전자전산학과 (공학박사)
- 2002년 9월 ~ 현재 : 금오공과대학교 컴퓨터공학부 조교수
- 관심분야: 컴퓨터 네트워크, 무선 센서 네트워크, 모바일 인터넷