

서브네트 ATM 모델에서의 UBR 연결 TCP 트래픽 전송을 위한 FB 기법

正會員 金 佑 駿*, 李 秉 基*

The FB Mechanism for TCP Traffic over UBR Connections in the Subnet ATM Model

Woo June Kim*, Byung Gi Lee* *Regular Members*

* 본 논문은 HAN/B-ISDN 단말 분야 광대역 단말기 개발 과제의 일환으로 수행되었음.

요 약

본 논문에서는 서브네트 ATM 모델에서 UBR 연결을 이용한 TCP 트래픽 전송 효율을 높일 수 있는 공평 버퍼 관리 (FB : Fair Buffering) 기법을 제안한다. 이 FB 기법을 이용하면 TCP 전송 수율 및 공평성이 모두 향상될 수 있음을 보인다. FB 기법은 UBR 상에서 TCP 전송이 가장 효율적으로 이루어지기 위해서는 버퍼 크기가 각 연결의 대역폭·지연 곱의 값에 비례해서 할당돼야 한다는 사실에 토대를 두고 있다. 기존의 최종 패킷 폐기(drop-tail), EB (Equal Buffering)과 제안된 FB 버퍼 관리 기법을 각각 RR (Round Robin) 스케줄링과 결합시켜 모의 실험하여 성능을 비교한 결과, FB 기법을 RR 스케줄링과 함께 사용할 때 그 성능이 가장 우수함을 보였다.

Abstract

In this paper, we present the FB(Fair Buffering) mechanism for the efficient support of TCP traffic over UBR connections in the subnet ATM model. We show that both throughput and fairness may be improved with this FB mechanism. The FB mechanism is founded on the observation that the performance of TCP over UBR connections is optimal when the buffer space is allocated in proportion to the connection's bandwidth-delay product. We compare through simulation the performance of the existing drop-tail, EB(Equal Buffering) and the proposed FB buffer management schemes, with and without RR(Round-Robin) scheduling, and demonstrate that the proposed FB mechanism is most effective when used in conjunction with the RR scheduling scheme.

* Lucent Technologies Hill, NJ., 연구원

* 서울대학교 전기공학부

論文番號 : 97384-1024

接受日字 : 1997年 10月 24日

I. 서 론

ATM 망이 전체 연결 경로 중 일부분만을 형성하는 서브네트 ATM 모델(subnet ATM model)에 있어서 TCP 전송은 여러 가지 요소의 영향을 받는다. 하나는 연결(connection) 자체의 성질이며, 이것은 사용되는 ATM 연결 종류 그리고 ATM 교환기 및 관문(gateway)에서 사용되는 체증 제어(congestion control) 기법에 의해서 결정된다. TCP 트래픽 전송을 위해 사용될 수 있는 ATM 연결 종류로는 UBR 서비스 및 최근에 정의된 ABR 서비스가 있다[1]. 본 논문에서는, UBR 연결을 사용하는 경우, 교환기 및 관문에서 사용할 가능한 체증 제어 기법에 대해서 고찰하고자 한다.

UBR 연결을 이용한 TCP 전송에서 제일 중요한 특징은 TCP 분절(segment)을 ATM 셀로 분할해서 전송하는 경우 성능이 상당히 낮아진다는 점이다. 셀 손실은 곧 TCP 분절의 손실로 이어질 수 있으므로, 비록 셀 손실률이 작더라도, 종단에 있는 사용자가 느끼는 데이터 손실률은 매우 클 수 있으며, 결과적으로 사용자 수율이 떨어질 수 있다. 또 소속 셀들 중 일부가 손실된 경우에도 나머지 셀들은 종단까지 계속 전송되므로 링크 대역폭을 낭비하게 된다. 이것을 일반적으로 분할 효과(fragmentation effect)라고 부른다 [2][3].

또 다른 영향을 주는 요소는 TCP내에 사용되는 TCP 체증 제어 기법이다[4][5]. TCP 체증 제어 기법에는 두 가지 문제가 있다. 첫째, 긴 경로의 연결들에 대해서 차별한다는 점이다. 즉, 여러 개의 연결들이 같은 체증점을 지나는 경우에, 경로가 짧은 연결들이 경로가 긴 연결들보다 더 많은 대역폭을 차지하게 된다. 둘째, 여러 개의 체증된 교환기 또는 관문을 지나는 경로를 갖는 연결들에 대해서 차별한다는 점이다. 즉, 두개의 연결들이 같은 지연을 갖더라도, 더 적은 수의 체증된 관문을 지나는 연결이 더 많은 대역폭을 차지하게 된다. ATM이 백본망으로 사용되는 예가 증가함에 따라 이러한 차별 대우가 문제를 일으킬 수 있는 상황이 현재 보다 더 자주 발생하게 된다.

본 논문에서는 서브네트 ATM 모델에서 UBR 연결을 사용할 경우에 대해 TCP 트래픽의 수율과 공평성을 높일 수 있는 방안에 대해서 연구를 하고자 한다. 새로운 버퍼 관리 기법과 단순한 스케줄링 방식을 결합시켜 사용하면 이 문제를 해결할 수 있음을 보이고, 또 다양한 실험 망구조에서의 실험을 통해 제안된 방안의 우수성을 보이도록 하겠다.

본 논문의 구성은 다음과 같다. 우선 2절에서는 ATM 망에서 UBR 연결을 이용한 TCP 트래픽 전송에 대해서 살펴보고, 이것을 바탕으로 3절에서 새로운 체증 제어 기법을 제안한 후, 끝으로 4절에서 모의 실험을 통해 제안된 기법의 성능을 분석한다.

II. 미래 ATM 망에서 TCP 트래픽 전송에 대한 고려 사항

미래의 공중 데이터망은 자원의 효율적인 사용과 동시에 사용자간의 격리(isolation)를 요구하게 된다 [6]. 즉, 사용자들은 다른 사용자에게 악영향을 미치지 않는 범위 내에서 최대한 높은 수율을 받을 수 있도록 해야 한다. 그러나, 기존의 체증 제어 기법들은 대부분 망 자원 사용의 효율화에 그 목표가 있었으며, 또 대부분의 사용자들은 협조적인 사용자였으므로 사용자들을 격리시키는 방안에 대한 연구는 드물었다.) 버퍼 관리는 많이 연구되었지만, 자원의 효율적인 사용과 관련된 연구 위주였으며, 사용자 격리를 목표로 하지는 않았다. 최근에 이르러, 다양한 QOS 및 서비스를 제공해야 종합 통신망이 대두되면서 사용자 격리 문제에 대한 연구가 산발적으로 진행되고 있을 뿐이다.

이러한 추세와는 별도로 ATM 고유 특성 때문에 생기는 제약도 있다. ATM의 가장 중요한 문제는 전송 단위(transmission semantics)와 제어 단위(control semantics)가 다르다는 것이다. 즉, 기본 전송 단위인 ATM 셀은 최적 제어 단위인 패킷 또는 프레임과는 다른 경우가 많다. 그래서 ATM 교환기에서는 초기 패킷 폐기(EPD : Early Packet Discard)와 같은 패킷

1) 예를 들어, TCP 체증 제어 기법은 체증이 일어나서 라우터에서 패킷이 폐기되면, TCP 신호원이 자발적으로 전송 윈도우를 줄인다. 이러한 체증 제어 기법은 신호원이 협조적이라고 가정하는 셈이다.

기반 제어 기법을 사용해야 그 손실 성능이 좋아진다 [2]. 또 전송 단위와 제어 단위의 불일치로 인해 ATM 망에서도 패킷 또는 프레임 단위의 귀환 정보 (feedback) 기법을 적용하기 어렵게 된다. 이것이 특히 문제되는 것은 비디오, 데이터 등 여러 가지 응용에 있어서 패킷 또는 프레임 단위의 귀환 정보가 가장 효과적이기 때문이다. 결과적으로 ATM 망에서의 체증 제어 기법은 주로 교환기 또는 관문에서 사용 가능한 스케줄링 또는 버퍼 관리 기법 위주로 해야 한다는 결론에 이르게 된다.

미래의 공중 데이터망의 요구 사항을 고려하면 데이터 트래픽을 위한 미래의 체증 제어 기법은 크게 스케줄링, 버퍼 관리, 귀환 정보, 종단 호스트 조정 (end-adjustment) 기법 등 네 가지로 구분할 수 있다 [6]. 그러나 위의 논의를 고려하여 공평하고 효율적인 서비스를 제공하려면 이들 네 가지 체증 제어 기법들 중에서 교환기 또는 관문에 바탕을 둔 스케줄링과 버퍼 관리 기법들을 사용하는 것이 바람직하다.

그러므로, UBR 연결을 이용한 TCP 연결들의 성능을 향상시키기 위해서는 결국 다음 네 가지 체증 제어 방법들이 가능함을 알 수 있다. 첫째, 특별한 버퍼 관리 기법이나 스케줄링 기법을 사용하지 않는 방법이다. 즉, 최종 패킷 폐기 기법과 FIFO 스케줄링을 이용하는 것이다. 이 경우에는 분할 효과 때문에 그 성능이 매우 떨어진다는 점은 이미 언급한 바 있다. 둘째, FIFO 스케줄링과 어느 한 가지 종류의 버퍼 관리 기법을 사용하는 방법이다. ATM에서 대표적인 버퍼 관리 기법은 EPD이지만, EB(Equal Buffering) 기법이나 RED(Random Early Detection) 기법과 같이 더 복잡한 버퍼 관리 기법을 사용하는 것도 가능하다. 셋째, 최종 패킷 폐기 기법과 어느 한가지 종류의 스케줄링 기법을 이용하는 방법이다. RR(Round-Robin)과 FQ(Fair Queuing)가 그 대표적인 예이다. 스케줄링을 사용함에 있어서 중요한 점은 UBR은 기본적으로 "최선 노력(best-effort)" 서비스이므로 복잡한 자원 예약 연결 설정 방식을 취하지 않는다는 점이다. 따라서 군집 크기, 예상되는 전송률 값 등에 의존하는 복잡한 스케줄링 기법에 의존할 수는 없다. 넷째, 버퍼 관리 기법과 스케줄링을 함께 사용하는 방법이다. 이 방법은 성능이 좋아지는 반면에 구현이 복잡할 가능성이 있다. 따

라서 구현이 간단하면서 성능이 우수하도록 만드는 방안을 강구해야 한다. 이 조합에 관해서는 다음절에서 더 자세히 고찰하도록 하겠다.

III. FB 기법과 UBR+FB+RR

이상적인 교환기는 모든 연결에 자원을 공평하게 분배해야 하며, 이때 자원의 공평 분배는 기본적으로 모든 연결들이 "최대-최소 공평(max-min fair)"한 대역폭을 받는 것으로 정의한다[1,10]. 그렇다면 이 경우에는 각 연결에 구체적으로 최소한의 자원이 할당되어야 한다. 분배할 자원은 대역폭과 버퍼 공간 두 종류이지만 두 개 중에서 버퍼 공간은 종속 변수이다. 즉, 연결들이 공평한 대역폭을 사용할 수 있도록 버퍼 공간이 할당되어야 한다.

이상적인 교환기/관문의 모델은 그림 1(a)과 같다. 그림에서 μ_i 와 B_i 는 각각 연결 i 의 대역폭 및 버퍼 공간을, μ 와 B 는 전체 대역폭 및 버퍼 공간을 나타낸다. 연결 i 에 할당된 버퍼 공간 B_i 는 그 연결의 최대-최

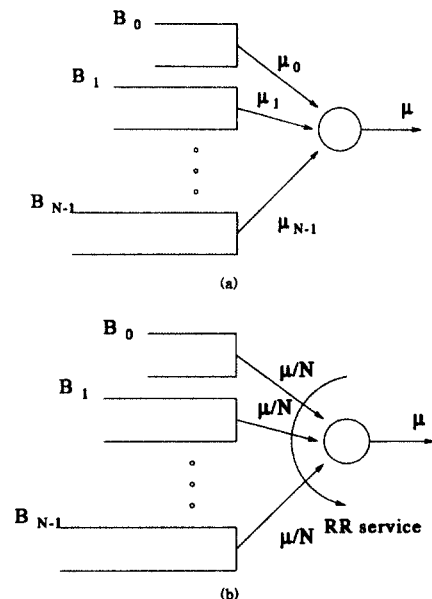


그림 1. 교환기/관문 출력 포트 모델 : (a) 이상적인 경우 : μ_i 과 B_i 는 i 연결의 대역폭 및 버퍼 공간이고 $\mu = \sum_{i=0}^{N-1} \mu_i$ 는 링크 대역폭. (b) 구현 가능한 경우 (FB+RR기법) : μ/N 은 각 연결의 대역폭, μ 는 링크 대역폭.

소 공평한 대역폭 μ_i 를 완전히 활용할 수 있는 크기의 공간이다. 여기서 문제는 μ_i 를 최대-최소 공평 분배에 의해서 정의할 때, B_i 의 값을 어떻게 정해야 하는가 하는 것이다.

최근 참고문헌 [7]에 소개된 분석을 토대로 이 문제에 대한 해답을 찾아보자. 이 분석에서는 TCP Reno 트래픽을 고정된 대역폭의 연결을 통해 전송할 경우, 버퍼 크기 B , 왕복 시간 RTT τ , 대역폭 μ 그리고 수율 λ 사이에 $\alpha = \frac{3(\beta+1)^2}{\beta^2+\beta+1}$ 의 관계가 성립함을 보였다.

단, $\alpha (= \lambda/\mu x)$ 는 실제 사용자 수율과 할당된 대역폭의 비이고, $\beta (= B/\mu\tau)$ 는 대역폭 · 지연의 곱으로 정규화시킨 버퍼 크기이다. 이 관계식의 주요 특징은 α 가 정규화된 버퍼 크기 $\beta (= B/\mu\tau)$ 에만 의존한다는 점이다. 이것은 대역폭이 정해져 있을 경우, 수율은 오직 버퍼 크기에 의해서 결정된다는 것을 뜻한다. 이 수식을 잘 검토하면 버퍼 크기가 대역폭 · 지연 곱만큼 될 때 링크 효율이 100%에 접근하게 됨을 알 수 있다. 따라서 개별 TCP 연결들이 정해진 대역폭을 할당받고, 또 각 연결에 해당 대역폭 · 지연 곱에 해당하는 버퍼가 할당될 때 그 할당된 대역폭을 완전히 사용할 수 있게 된다. 이와 같은 분석을 바탕으로 참고 문헌 [7]에서는 ABR 연결상의 TCP 전송 성능 향상 방안을 제시했고, 본 절에서는 이 분석을 UBR 연결상의 TCP 전송에 대해 적용하여, TCP 연결의 성능을 향상시킬 수 있는 방안을 제시하도록 하겠다.

공평한 대역폭 할당은 대역폭을 모든 연결에 균등하게 할당하는 것으로 근사시킬 수 있다. 즉, N 개의 연결이 있고 링크 대역폭이 μ 이면 각 연결에 대해 μ/N 씩 할당하도록 한다. 균등한 대역폭 분배는 최대-최소 공평한 대역폭 할당과 일치하지는 않지만, 그 근사적인 효과를 낼 수 있다. 대역폭 할당이 완전히 공평하게 되지 못하는 점은 버퍼 공간을 공평하게 할당하는 FB(Fair Buffering) 기법을 버퍼 관리 기법으로 사용함으로써 보완할 수 있다. 여기서 버퍼 공간을 공평하게 할당한다는 것은 모든 연결들에게 같은 크기의 버퍼 공간을 할당하는 균등한 분배를 의미하는 것이 아니라, 각 연결이 최대-최소 공평한 대역폭을 활용할 수 있게 해주는 양만큼을 할당한다는 뜻이다.

FB 기법은 기본적으로 다음 두 가지 관점에 바탕을

두고 있다. 우선, 앞에서 논의했듯이 고정된 대역폭으로 서비스를 받는 TCP 연결의 수율은 정규화된 버퍼 β 에 의해서 결정된다는 점이다. 즉, 각 연결이 할당된 대역폭을 공평하게 사용할 수 있기 위해서는 각 연결에 할당된 버퍼 크기가 대역폭 · 지연 곱 값에 비례해야 하는 것이다. 둘째는, TCP 체증 제어 기법은 패킷 손실이 균집을 이루지 않고 고루 퍼져 있을 때 그 성능이 가장 좋다는 점이다. 그런데 TCP Reno의 성능은 체증 기피 주기(congestion-avoidance cycle)동안 한 개의 패킷 손실이 있을 때 가장 우수한 것으로 알려져 있으므로[2], 체증 기피 주기 동안 전송되는 패킷 수가 N_{ca} 이면, 손실 패킷 사이에 N_{ca} 개의 패킷이 전송될 때 성능이 가장 좋아지는 것을 알 수 있다.

이와 같은 FB 버퍼 관리를 RR 스케줄링과 결합해서 UBR 트래픽에 적용하면 위에서 검토한 이상적인 경우에 대한 근접한 성능을 얻을 수 있게 된다. 편의상 이 조합을 UBR+FB+RR 기법이라고 부르기로 하자. 이 기법이 우수한 성능을 보이게 되는 것은 각 연결의 수율이 그 연결에 실제로 할당된 버퍼 크기와 대역폭 · 지연 곱의 비에 의해 결정된다는 사실을 실제 적용하고 있기 때문이다. 즉, RR 스케줄링은 각 연결에 균등하게 서비스 기회를 주지만, FB 기법을 통해 버퍼 공간을 차별적으로 할당함으로써, 각 연결이 결과적으로 최대-최소 공평 분배와 같은 대역폭을 사용할 수 있도록 하는 것이다.

UBR+FB+RR 기법은 그림 2에 보인 것과 같이 네 가지 단계로 구성된다. 먼저, 첫째 단계에서 각종 변수들을 초기화하고, 둘째 단계에서 대역폭의 평균을 계산하며, 셋째 단계에서는 대역폭 · 지연 값의 변화에 따라 달라지는 변수들을 새로 계산하고, 넷째 단계에서는 각 패킷마다 폐기여부를 결정한다.

이 기법을 설계하는데 있어서 다음과 같은 여러 가지 사항들이 고려되었다. 우선, 망 상태의 변화에 의한 요동을 최소화하고 또한 계산량을 줄이기 위해서 큐 문턱값 $QThresh_i$ 와 주기 문턱값 $WThresh_i$ 는 $N\beta$ 패킷마다 계산하도록 한다. 버퍼 크기를 대역폭 · 지연 곱의 값에 비례해서 할당해야한다는 점은 $QThresh_i$ 에 반영되어 있고, 한 체증 기피 주기 동안에는 한 개의 패킷만 잃도록 해야 한다는 점은 $WThresh_i$ 에 반영되어 있다. $WThresh_i$ 의 값으로는 참고 문헌 [7]에

```

1) 초기화
   FirstCyclei = TRUE ;
   NumPkti = 0 ;
   QThreshi = BSize ;
   WThreshi = BSize ;

2) 연결 설정 또는 해제
   BwAvgi = BwPeak / N ;
   BwDlyi = BwAvg × RTTi ;

3) Nth 패킷을 전송할 때마다
   if FirstCyclei is TRUE
     QThreshi = InitThresh / 4 ;
     WThreshi = BwDlyi ;
   else
     QThreshi = InitThresh / 4 ;
     WThreshi = max ( 2, BwDlyi / 4 ) × BwDlyi ;

4) 전송할 패킷을 받을 때마다
   if QLeni > QThreshi and NumPkti > WThreshi
     drop packet ;
     NumPkti = 0 ;
     FirstCyclei = FALSE ;
   else
     NumPkti = NumPkti + 1 ;
    
```

그림 2. FB기법 (BSize: 최대 버퍼 크기, BwAvg_i: 연결의 예측된 대역폭, BwPeak: 링크 최대 대역폭, N: 연결 수, RTT_i: ATM 연결의 지연 시간, QLen_i: 연결의 큐 길이, NumPkt_i: 최근 손실 패킷 이후 전송된 패킷 수)

Fig. 2. FB Mechanism (Bsize: Maximum buffer size, BwAvg_i: estimated bandwidth of connection, Bwpeak: maximum link bandwidth, N: number of connection, RTT_i: round trip time of connections, QLen_i: queue length of connection, NumPkt_i: number of packets transmitted since last dropped packet)

계산된 Nca의 하한값인 BwDly_i/4을 이용한다.

패킷 손실을 분산시키기 위해서 패킷 손실후 관문을 지나가는 패킷 수를 계속 기록한다. 지나간 패킷 수가 WThresh_i 보다 커지면 하나의 체증 기피 주기가 끝났다고 생각하고, 만약 이 때 큐의 길이가 QThresh_i 보다 크면 패킷을 폐기한다. TCP 연결이 맨 처음 시작시 거치는 느린-시작 (slow-start) 주기 단계에서의 기하급수적 윈도우 확장 때문에 생길 수 있는 패킷 군집과 그 영향으로 생기는 군집성 손실에 의한 성능 저하를 막기 위해서 연결 초기에는 QThresh_i 및 WThresh_i의 값을 보통 때보다 작은 값으로 설정한다. 연결 초기인지 아닌지는 FirstCycle_i 변수를 통해서 안다. 이 방법을 사용함으로써 기하급수적인 윈도우 확장이 군집성 패킷 손실을 일으키기 전에 단일 패킷

손실을 일으켜, TCP 윈도우가 원활하게 커지고 타임아웃 등이 발생하지 않도록 한다.

FB 기법은 기본적으로 대역폭·지연 곱의 값에 바탕을 두고 있다. 연결에 할당되어 있는 대역폭이 정해져 있지 않고 연결의 지연도 고정되어 있지 않은 전통적인 “최선 노력” 망에서는 대역폭·지연 곱을 계산하는 것이 불가능했다. 그러나 ATM은 기본적으로 연결성 망 프로토콜이므로 연결 설정 과정에서 각 연결의 고정 지연은 알게 된다. 또 RR 스케줄링 방식이 이용될 때는 각 연결에 링크 대역폭이 균등하게 할당되어 있다고 생각할 수 있다. 이 두 가지를 바탕으로 대역폭·지연 곱의 값을 계산할 수 있다. 물론 이 때 사용된 대역폭은 실제 사용 대역폭과는 다르고, 또 지연 값도 고정 지연 외에 큐잉에 의한 가변 지연도 있으므로 값이 정확하지는 않다. 그러나 다음절의 모의 실험 결과에서 알 수 있듯이 이것이 큰 영향을 미치지 않는다. 그 이유는 UBR+FB+RR 기법에서 주로 계산된 대역폭·지연 곱의 비를 이용할 뿐, 절대적인 값은 별로 사용되지 않기 때문인 것으로 볼 수 있다. 또 그 절대적인 값을 이용한다 하더라도 제안 기법에서 사용하는 계산 방식이나 변수 값들은 실질 값의 하한값에 해당하므로 그 성능을 악화시키는 방향으로 나아가지는 않게 된다.

IV. 모의 실험을 통한 UBR+FB+RR 성능 검증

1. 모의 실험 모델

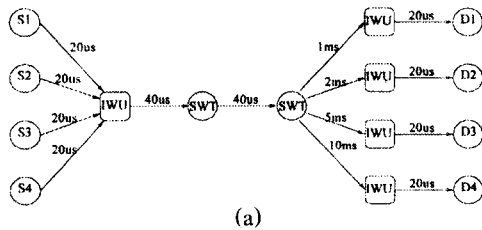
우선, ATM 망에서 TCP 전송에 관한 성능을 분석하기 위해 사용된 모의 실험 모델 및 실험 망구조들에 대해서 설명하겠다. 기본적인 모의 실험 도구는 REAL 망 모의 실험 라이브러리이다[8].

모의 실험에서 UBR+, UBR+EB, UBR+FB, UBR+RR, UBR+EB+RR, 그리고 UBR+FB+RR 등 여섯 개의 체증 제어 기법들을 비교했다. 처음 세 가지 기법은 FIFO 스케줄링을 이용했고, 나머지는 RR 스케줄링을 이용한 것이다. UBR+과 UBR+RR 기법은 최종 패킷 폐기 기법을 이용한다. EB가 이름에 포함되어 있는 기법들은 Li[9]의 버퍼 관리 기법을 사용하고, FB가 이름에 포함되어 있는 기법들은 위에서 제안된 FB 기법을 사용한 것이다. 모든 경우에 대해, 분

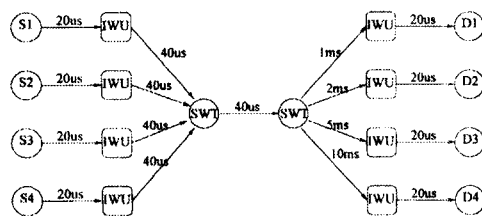
할 효과에 의한 성능 저하를 막기 위해서 버퍼가 어떤 경계값을 넘으면 최초로 도착하는 새 패킷의 셀들을 모두 폐기하는 것으로 했다. 이때 경계값은 최대 버퍼 크기에서 UBR 연결의 수와 최대 패킷 크기의 곱을 뺀 값으로 정했다.

TCP 신호원들은 무한 TCP Reno 신호원이고 윈도우 확장 선택 사양(window scaling option)을 사용했다. TCP 분절(segment)들의 길이는 1,500 바이트(B)였으며 ATM 적응 계층으로 AAL5를 사용했다. TCP 타이머의 정밀도는 50ms이고, 지연 ACK 선택 사양(delayed ACK option)은 사용하지 않았다[5].

모의 실험에서 사용한 실험 망구조는 그림 3(a)과 (b)에 나타난 것과 같다. 이들 망구조 A와 B는 각각 관문 및 교환기 체증 제어 기법이 서로 다른 지연을 갖는 TCP 연결들의 성능에 미치는 영향을 분석할 수 있도록 설정한 것이다. 그림에서 TCP 신호원 Si는 목적지 Di에 패킷을 전송한다. 각 링크의 대역폭은 50Mbps이다. 모든 관문과 교환기에 있는 버퍼들은 모두 크기가 같다.



(a)



(b)

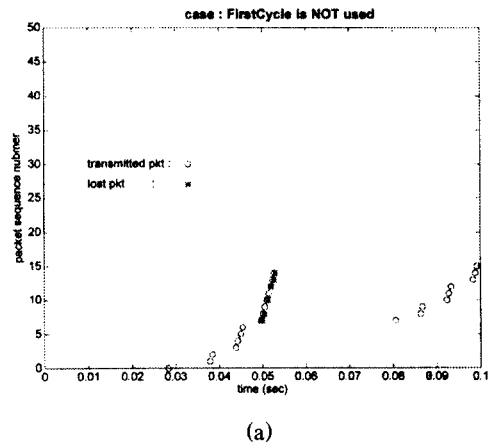
그림 3. 실험 망구조: (a) A, (b) B. (Si : TCP 신호원, Di : TCP 목적지, IWU : 관문, SWT : ATM 교환기)

Fig. 3. Test Topology : (a) A, (b) B. (Si : TCP destination, Di : TCP destination, IWU : gateway, SWT : ATM switch)

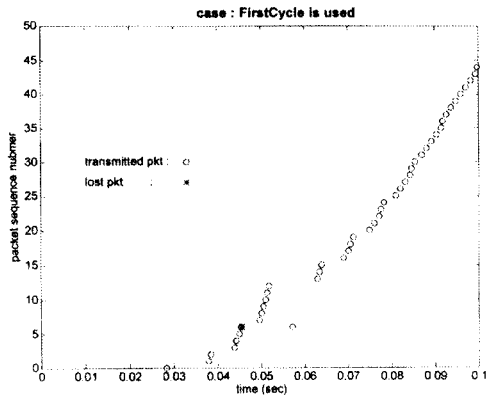
여러 가지 체증 제어 기법들의 성능은 수율 및 공평성 면에서 비교했다. 수율 지수로는 전체 수율을 대역폭이 최대-최소 공평하게 할당되어 있는 최대 수율로 정규화시킨 값을 이용했다. 이때 정규화된 수율이 1.0보다 큰 것은 대역폭 최대-최소 공평하게 할당되어 있지 않는 경우에 발생하며, 이때 짧은 연결들이 더 많은 대역폭을 차지하게 된다. 공평성 지수로는 Jain의 지수를 이용했다[11]. 이 값은 1에 가까울수록 더 공평하게 대역폭이 할당되어 있다는 것을 나타낸다. 버퍼 크기는 실험 망구조에서 최대 대역폭 및 최대 지연의 곱으로 정의되는 이상적인 최대 버퍼 크기에 대해서 정규화시켰다.

2 모의 실험 결과 및 분석

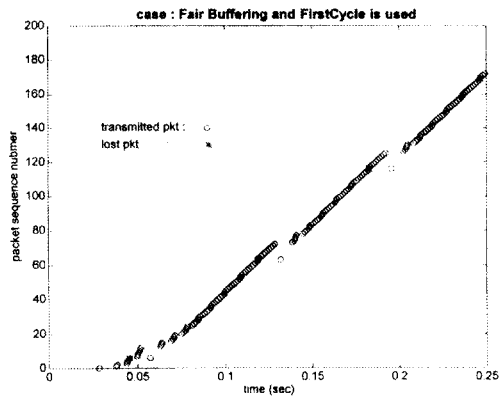
그림 4는 실험 망구조 A에서 정규화된 버퍼 크기가 0.25인 상태일 때 FB 기법의 동작을 확인한 실험 결과이다. 그림은 S3 - D3 연결에 대한 전송 패턴을 나타낸 것이다. 그림 4(a)와 (b)를 비교하면 FirstCycle 변수의 효과를 알 수 있다. 그림 4(a)는 변수를 사용하지 않은 경우로서, 느린 시작(slow-start) 구간에서 TCP 윈도우가 기하급수적으로 확장되므로 군집성 패킷 손실 및 타임아웃이 일어나게 되는 것을 볼 수 있다. 이와 반대로 그림 4(b)는 FirstCycle 변수를 사용한 경우로서, 큐가 너무 커지기 전에 패킷 하나를 폐기하므로써 윈도우가 원활하게 확장되는 것을 볼 수 있다. 그림 4(c)는 FirstCycle 변수를 포함하는 FB 기법의 전체적



(a)



(b)



(c)

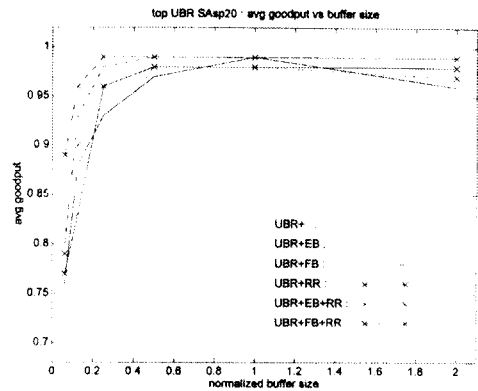
그림 4. UBR+FB+RR 동작 확인 실험 : (a) FirstCycle 변수를 사용하지 않는 경우, (b) FirstCycle 변수를 사용한 경우, (c) FB 기법의 경우(FirstCycle 변수포함).

Fig. 4. UBR+FB+RR operation : (a) When the FirstCycle variable is used, (b) when the FirstCycle variable is not used, (c) the operation of the whole FB mechanism including the use of the FirstCycle variable.

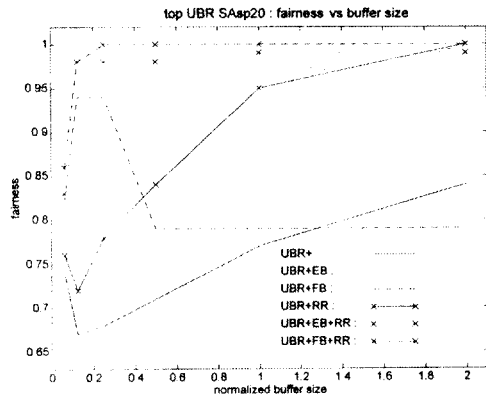
인 동작을 보인 것이다. 의도했던 대로 한 체증 기피 주기 동안에는 단일 패킷손실만 일어나고, 따라서 패킷 전송이 규칙적으로 이루어지는 것을 볼 수 있다.

그림 5(a)와 (b)는 실험 망구조 A 상에서 단일 체증 점이 관문인 경우에 여러 기법들의 수율과 공평성을 비교한 것이다. 우선 버퍼 관리 기법의 영향을 고려하기 위해서 UBR+, UBR+EB, UBR+FB 기법들을 비교하면, 버퍼 크기가 클 때는 어느 버퍼 관리 기법을

사용하더라도 수율 성능이 우수하다는 것을 알 수 있다. 그러나 버퍼가 작을 때에는 FB 기법(즉, UBR+FB)이 가장 우수하고 이에 비해 최종 패킷 폐기 기법(즉, UBR+)과 EB 기법(즉, UBR+EB)의 성능이 떨어지는 것을 볼 수 있다. 공평성 측면에서는 EB 또 FB 기법 모두가 공평성을 향상시킨다. 그러나 EB 기법의 공평성은 버퍼가 커지면서 좋아지는 반면, FB 기법은 버퍼 크기가 커지면 오히려 공평성이 나빠진다. 이것은 큐내의 패킷 분포가 균일하지 않아서 생기는 HOL (Head of Line) 블로킹 현상 때문이다. EB 기법은 HOL 현상의 영향을 덜 받아서 그 성능이 떨어지지 않는다. 이 HOL 블로킹 문제는 FIFO 이외의 다른 스



(a)



(b)

그림 5. 실험 망구조 A 모의 실험 결과 : (a) 전체 수율, (b) 공평성.

Fig.5. Simulation results for test topology A : (a) Total goodput, (b) fairness.

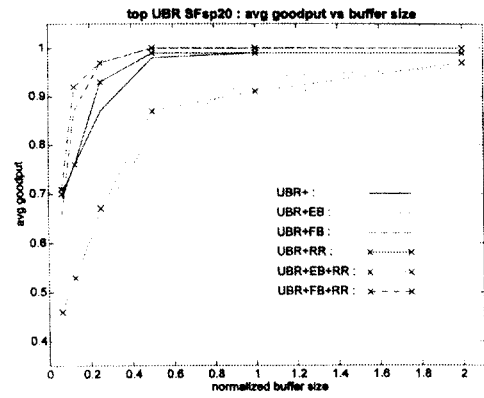
케줄링 방식을 이용하면 해결할 수 있게 된다. 이것은 다음 RR 스케줄링을 결합한 경우에 볼 수 있다.

위의 3 가지 경우에 RR 스케줄링과 버퍼 관리 기법을 결합해서 사용하면 버퍼 크기가 클 때에는 모든 기법들이 FIFO 스케줄링을 이용한 경우와 마찬가지로 우수한 수율을 갖는다. 그러나 버퍼가 작을 때에는 FB 기법(즉, UBR+FB+RR)의 수율이 제일 좋고, EB 기법(즉, UBR+EB+RR)과 최종 패킷 폐기 기법(즉, UBR+RR)의 성능은 떨어지는 것을 알 수 있다. 그러나 FIFO 스케줄링을 이용했던 경우와는 달리, 모든 버퍼 크기에서 FB 기법이 EB 또는 최종 패킷 폐기에 비해 공평성이 우수하다. 이것은 FIFO 스케줄링을 사용했을 때 나타났던 HOL 블로킹 현상이 RR 스케줄링을 사용함으로써 해결된 것을 의미한다.

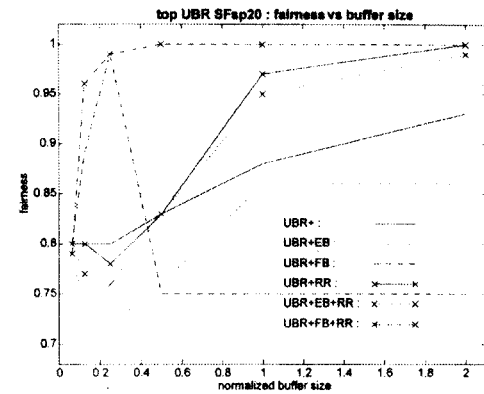
그림 6은 실험 망구조 B상에서 단일 체증 점이 교환기인 경우에 대한 실험 결과들을 나타낸 것이다. 전체적으로는 실험 망구조 A의 경우와 유사한 경향을 나타내지만, FB 기법과 EB 기법간의 성능차가 더 커지는 것이 특징이다. 즉, 단일 체증 점이 관문이 아니라 ATM 교환기일 때에는 FB 기법이 상대적으로 더 우수한 성능을 나타내게 되는 것이다.

이 실험결과를 종합하면 그림 5와 6 모든 경우에 대해서 UBR+FB+RR 기법이 가장 우수한 것으로 나타났다. UBR+EB+RR 기법은 관문이 유일한 체증점일 경우에는 UBR+FB+RR과 거의 비슷한 성능을 나타내지만, 단일 체증점이 ATM 교환기일 경우에는 그 성능이 많이 떨어진다. 이에 비해, UBR+FB+RR 기법을 사용하면 이를 두 가지 경우에 대해 수율 및 공평성 모두 향상되게 된다. 이것은 FB 기법이 기본적으로 짧은 연결들이 버퍼를 독점하는 것을 막아 긴 경로의 연결들이 할당 대역폭을 완전히 사용할 수 있도록 해주기 때문에 가능해지는 것이다.

모의 실험 결과에 의하면 UBR+FB+RR 알고리즘은 정규화된 버퍼 크기가 작을 때 더욱 효과적인 것으로 나타난다. 망구조상에서 최대 대역폭 및 최대 지연의 곱으로 정의된 이상적인 최대 버퍼 크기로 정규화했다는 점을 상기하면, 정규화된 버퍼 크기가 작을 경우가 WAN에서는 더 일반적이 되는 것을 알 수 있다. 그러므로 이러한 WAN 환경에서는 UBR+FB+RR를 사용하면 성능이 상당히 향상될 것으로 기대된다.



(a)



(b)

그림 6. 실험 망구조 B 모의 실험 결과 : (a) 전체 수율 대 버퍼 크기, (b) 공평성 대 버퍼 크기.

Fig. 6. Simulation results for test topology B : (a) Total goodput, (b) fairness.

RR 스케줄링의 사용 유무는 전체 성능에 큰 영향을 미치는 것으로 파악됐다. 특히, FB 기법을 사용할 경우, FIFO 스케줄링을 이용할 경우에는 HOL 블로킹 현상으로 인해 버퍼가 커지면서 공평성 성질이 크게 떨어지지만, RR 스케줄링을 사용하면 이 문제가 해결되는 것을 볼 수 있었다. 또 RR 스케줄링을 사용할 경우에는, 어느 버퍼 관리 기법과 결합하더라도 충분히 큰 버퍼를 할당하면 수율 및 공평성이 우수해지는 것을 볼 수 있다. 그러나 이 경우에 정규화된 버퍼가 크다는 것은 WAN 규모의 망에서는 실질적으로 필요한 버퍼가 대단히 큰 것을 의미함을 유념해야 한다.

