

論文2002-39TC-12-2

ATM-GFR에서 최소 전송율 보장 및 공정성 향상을 위한 Leaky Bucket 기반의 버퍼 관리 알고리즘

(Leaky Bucket Based Buffer Management Algorithm to Guarantee MCR and Improve Fairness in ATM-GFR)

金觀雄*, 金辯坤**, 田炳實***

(Kwan-Woong Kim, Byun-Gon Kim, and Byoung-Sil Chon)

요약

최근 ATM 포럼에서 새로운 서비스 클래스인 GFR을 제안하였다. GFR 서비스는 TCP/IP 와 같은 "best-effort" 트래픽을 서비스하기 위해서 고안되었다. GFR 서비스는 VC의 최소 수율을 보장하고, 잉여 대역을 연결된 VC에 공정히 제공하여야 한다. 본 논문에서는 GFR이 요구하는 최소 전송율 보장과 공정성을 제공하기 위해서 Leaky-bucket기반의 버퍼 관리 알고리즘을 제안한다. 제안된 버퍼 관리 알고리즘은 실제 구현이 용이하도록 Leaky-bucket의 프로세싱 부하와 구현 복잡도를 줄였다.

Abstract

The ATM Forum recently introduced the Guaranteed Frame Rate(GFR) service category. GFR service has been designed to support classical best effort traffic such as TCP/IP based traffic. The GFR service not only guarantees a minimum throughput, but also supports fair distribution of available bandwidth to completing VCs. In this paper, we proposed a new buffer management algorithm based on leaky bucket to provide minimum cell rate guarantee and improve fairness. The proposed algorithm reduces complexity and processing overhead of leaky-bucket algorithm to implement easily.

Keywords : Buffer management, GFR, TCP/IP, ATM

* 正會員, 韓國標準科學研究院 人間情報그룹 Post. Doc.

(Korea Research Institute of Standards and Science)

** 正會員, 群山大學校 工科大学 電子情報工學部 BK

(Division of Electronic and Information Eng., Kunsan Nat'l Univ.)

*** 正會員, 全北大學校 工科大学 電子情報工學部

(Division of Electronic and Information Eng., Chonbuk Nat'l Univ.)

接受日字:2002年7月31日, 수정완료일:2002年11月25日

I. 서론

최근 인터넷의 광범위한 보급으로 인해 기존의 ATM 네트워크에서 인터넷 트래픽을 효율적으로 서비스할 필요성이 요구되었다. ATM 포럼에서는 인터넷 트래픽과 같은 비실시간성 트래픽을 서비스하기 위해 GFR 서비스 클래스를 제안하였다^[1]. GFR 서비스는 UBR 서비스에 최소의 서비스 품질 보장을 해주기 위한 방식으로 UBR+라는 이름으로 제안되어 연구가 진행되다가 GFR로 바뀌어 현재 ATM Forum TM 4.1에서 표준화가 되

었다. GFR은 연결된 VC에 대해 MCR(Minimum Cell Rate)를 보장하고 망에서 사용하고 남은 여분 대역폭을 각 VC에 공평하게 할당해야 한다.

R. Goyal는 GFR에서는 세 가지 기법을 통해서 MCR을 보장할 수 있다고 제안을 했다^[2]. 세 가지 기법은 policing, buffer management, scheduling이다. Policing은 셀 레벨에 있어서 보장을 프레임 레벨에서 보장할 수 있도록 맵핑시키는 기능을 담당하고 프레임 단위로 셀을 순응시키기 위해 F-GCRA(Frame based Generic Cell Rate Algorithm)를 사용한다. 버퍼 관리 기법은 각 VC의 버퍼 점유를 관리하기 위해 사용된다. 스케줄링 기법은 프레임들이 어떠한 방식으로 다음 hop으로 스케줄링하는데 사용된다. R. Guerin과 J. Heinanen는 버퍼 관리를 위한 두 가지 구현방식을 제안하였다^[3, 4]. 하나는 perVC-Queueing과 perVC-accounting이고 다른 하나는 FIFO 큐잉과 per-VC policing을 사용한 기법이다. TCP/IP 트래픽과 같은 버스티한 특성의 트래픽을 서비스할 경우 FIFO 큐잉과 perVC-policing 방식은 GFR이 요구하는 최소 전송을 보장과 공평성을 제공하기 어려운 단점이 있다^[1, 3]. perVC-Queueing 방식은 연결된 VC에 대해 버퍼를 할당하므로 MCR과 잉여 대역의 공평 할당이 쉬운 장점이 있으나 모든 VC에 논리적인 큐를 할당하고 출력단에 스케줄러가 필요하므로 하드웨어 구현과 프로세싱 복잡도가 높은 방식이다.

GFR 서비스를 통해서 TCP 트래픽을 전송할 때 TCP 연결 간의 RTT(Round Trip Time)과 MSS(Maximum Segment Size)등의 파라미터는 VC의 수율과 공평성에 크게 영향을 미친다. 기존의 FIFO 기반의 버퍼 관리 알고리즘은 연결된 VC의 전송 지연시간이 다를 경우 혹은 패킷의 크기가 다르게 설정되었을 경우 VC가 요구하는 MCR 보장 및 여분 대역의 공평한 할당이 어렵다. 본 논문에서는 Leaky-bucket 기반의 버퍼 관리 알고리즘을 제안하고 컴퓨터 시뮬레이션을 통하여 기존의 버퍼 관리 알고리즘과 성능을 비교 검토하였다.

II. 기존 버퍼 관리 알고리즘

GFR 서비스는 VC에 예약된 MCR을 보장하고 여분의 대역을 공평히 할당할 수 있어야 한다. GFR에서 버퍼 관리는 네트워크 대역을 효율적으로 사용하고 폭주상태를 해결하는 기능 뿐 아니라, VC별 데이터 트래픽의 수율을 제어하여 VC에 할당된 MCR을 보장하고 공평성을

향상시키는 기능을 수행한다. GFR에서 TCP 성능을 향상시키기 위해서 Double-EPD(Double-Early Packet Discard)^[3]와 DFBA(Differential Fair Buffer Allocation)^[5] WFBA(Weighted Fair Buffer Allocation)^[6]과 같은 FIFO 기반의 버퍼 관리 알고리즘이 ATM 네트워크에서 GFR를 지원하기 위해서 제안되었다.

1. Double-EPD

R. Guerin과 J. Heinanen이 제안한 Double-EPD는 프레임 태깅(GFR.2)과 FIFO 큐를 기반으로 간단한 버퍼 관리 알고리즘으로 LBO(Low Buffer Occupancy)와 HBO(High Buffer Occupancy), 두 개의 전역 임계치를 사용한다. FIFO 큐의 버퍼 점유율 X 가 LBO보다 크면 새로 도착하는 CLP(Cell Loss Priority) = 1인 패킷은 폐기된다. CLP = 0인 패킷의 경우 HBO 이하에서는 모두 큐로 받아들이고 X 가 HBO 이상이면 폐기하는 방식으로 실제 스위치에서 구현하기가 용이하나 각 연결별로 버퍼 관리가 안되므로 GFR이 요구하는 MCR 보장 및 여분 대역의 공평한 할당이 불가능한 방식이다.

2. DFBA

R. Goyal은 FIFO 큐를 사용하는 DFBA를 제안했다. DFBA는 Double-EPD 기반으로 두 개의 전역 임계치를 사용하며, 연결된 모든 VC에 동적 임계치를 제공하고 확률에 따른 패킷 폐기 방식을 사용하여 VC가 요구하는 MCR을 보장하는 방식이다^[5].

DFBA에서 패킷의 폐기여부를 결정하기 위해 Double-EPD 방식과 같은 두 개의 전역 임계치인 L 과 H 를 사용한다. 전체 버퍼 점유량 X 가 L 이상이 되면 CLP = 1인 프레임은 폐기되고, CLP = 0인 프레임은 해당 VC의 버퍼 점유율과 전체 버퍼 점유율 X 에 따른 식 (1)과 같은 확률 값에 따라 폐기된다. 이 확률 함수는 MCR에 따른 VC별 공평성을 향상시키고 버퍼 효율성을 개선하기 위한 부분으로 나타난다.

$$P_{drop} = Z, \left(\alpha \frac{X_i - X \times W_i / W}{X(1 - W_i / W)} + (1 - \alpha) \frac{X - L}{H - L} \right) \quad (1)$$

여기서, Z 는 스케일링 요소이며 α 는 두 항 사이에 비중을 조절하기 위한 요소이고 W_i 는 i -번째 VC의 가중치로 $W_i = MCR_i / \sum MCR_j$ 로 계산하고, $W = \sum MCR_i$ 이다.

3. WFBA

WFBA 버퍼 관리 알고리즘은 ATM 포럼에서 GFR

서비스 클래스가 연구되기 이전에 제안되었다^[6]. 그러나 WFBA는 ATM 스위치에서 GFR 서비스를 제공하기 위해 사용될 수 있으며 perVC-accounting에 기반으로 FIFO 기반 스위치에 적용될 수 있으며, 버퍼의 사용 가능한 공간을 활성 VC에 각 VC의 가중치에 비례하여 할당한다. WFBA는 K, R의 두 개의 버퍼 임계치 변수를 가지며, K는 EPD의 임계치와 같다. 버퍼 점유율(B_{occ})이 임계치 R 이하이면, 스위치는 폭주상태가 아니라고 판단하고 각 VC의 버퍼 이용율에 관계없이 도착하는 모든 패킷을 수용한다. 버퍼 점유율이 임계치 이상이면 도착하는 패킷은 VC의 현재 버퍼 점유율과 가중치에 따라 수용되거나 폐기된다. 이를 위해 WFBA는 각 VC의 버퍼의 셀 수를 저장하며 조정 파라미터 $Z(0 < Z < 1)$ 를 사용한다. 버퍼 점유율이 임계치 R을 초과하고 VC[i]의 새로운 패킷이 도착하는 경우, 수식 (2)가 참이면 받아들인다.

$$B_{occ}[i] \leq Z \times B_{occ} \times \left(\frac{MCR[i]}{\sum_{j=1}^{ActiveVCs} MCR[j]} \right) \times \left(1 + \frac{K - B_{occ}}{B_{occ} - R} \right) \quad (2)$$

WFBA는 GFR 트래픽 계약의 MBS(Maximum Burst Size) 파라미터를 지원하지 않으며 낮은 MCR이나 MCR이 0인 GFR VC를 지원하기 위한 문제가 있으므로 GFR서비스를 완전하게 지원하기 어렵다^[7].

III. 제안된 버퍼 관리 알고리즘

perVC-accounting 버퍼 구현 방식은 새로 도착하는 셀을 버퍼에 수용 여부를 제어하는 기법으로 perVC-Queueing 기법에 비해 구현이 용이하고 간단하나 TCP 트래픽의 버스트 특성으로 인해 GFR이 요구하는 최소 전송을 보장 및 높은 공평성을 보장하기가 어렵다. 특히 TCP 계층에서 RTT와 패킷의 크기가 다를 경우 즉, VC의 수율이 종단 간의 지연시간과 패킷의 크기에 따라 크게 영향을 받을 수 있어서 GFR 서비스를 통해 공평한 서비스가 어렵다. TCP의 트래픽 특성이 버스트한 원인은 슬라이딩 윈도우(sliding window)방식을 사용하고, 폭주 회피 및 복구를 위해서 폭주회피(congestion avoidance)와 느린 시작(slow start)알고리즘을 사용하기 때문이다^[8,9]. 이 두 가지 알고리즘은 패킷의 왕복시간인 RTT와 TCP의 최대 세그먼트 크기 MSS 파라미터에 매우 밀접하게 관련되어 있기 때문에 기존의 버퍼 관리

알고리즘을 사용할 경우, RTT가 짧은 연결이 긴 연결보다 서비스를 많이 받고, MSS가 큰 연결이 작은 연결보다 서비스를 많이 받는 현상이 발생한다.

위와 같은 문제점들을 해결하기 위해서 GFR VC를 통해 TCP 트래픽을 서비스할 경우, Leaky Bucket을 사용하면 VC의 MCR 보장과 공평성을 제공할 수 있다. 그러나 연결된 모든 VC에 Leaky Bucket을 할당하는 방식은 하드웨어 구현의 복잡성과 프로세싱 부하가 크므로 현실적인 구현방안으로는 적합하지 않다. 제안된 버퍼 관리 알고리즘은 Leaky Bucket 알고리즘을 기반으로 실제 구현이 용이하도록 계산 복잡도를 낮추고 단순화하였으며 Leaky Bucket의 장점을 그대로 수용하였다.

1. 셀 카운터 갱신 과정

제안된 버퍼 관리 알고리즘은 Leaky Bucket 알고리즘을 기반으로 구현 복잡도를 낮추기 위해서 각 VC에 Leaky Bucket을 할당하는 대신에 셀 카운터와 최근 카운터 갱신시간의 두 가지 파라미터를 할당하고, 패킷의 첫 번째 셀이 도착 할 때, 셀 카운터와 갱신시간을 계산한다. VC의 셀 카운터 파라미터는 일정 시간 내의 할당된 대역에 따라 서비스 받아야할 셀 수를 의미하며 셀 카운터로 패킷의 수용여부를 판단하여 VC가 할당받은 대역을 보장받을 수 있다. VC의 패킷의 첫 번째 셀이 도착할 때 토큰의 수를 식 (3)과 같이 계산한다. 토큰은 i-번째 VC의 패킷 도착시간 t_a 와 최근 갱신시간 $LT_i(k-1)$ 사이에 VC의 할당대역에 따라 발생하는 토큰 수를 의미한다.

$$Token_i = \left\lfloor \frac{t_a - LT_i(k-1)}{T_i} \right\rfloor \quad (3)$$

- 여기서 t_a : i-번째 VC의 k 번째 패킷의 첫 번째 셀이 도착 시간
- $LT_i(k-1)$: k-1 번째 패킷이 도착했을 때 토큰이 갱신된 시간
- T_i : i-번째 VC에 할당된 대역에 따른 셀 간격시간
- $\lfloor x \rfloor$: x보다 작은 최대 정수

식 (4)는 VC에 할당된 대역 $rate_i$ 를 계산하는 식이다. T_i 는 VC에 할당된 대역의 역으로 식 (5)와 같이 계산할

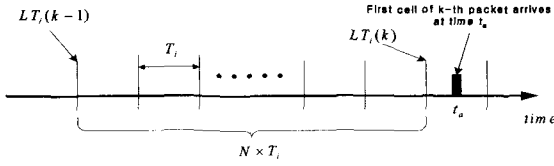


그림 1. 최근 토큰 갱신 시간 계산 과정
Fig. 1. Last token generation time procedure.

수 있다.

$$rate_i = MCR_i + \frac{C_{GFR} - \sum_{j=1}^N MCR_j}{N} \quad (4)$$

$$T_i = 1 / rate_i \quad (5)$$

여기서, MCR_i : i -번째 VC에 할당된 MCR
 C_{GFR} : GFR 클래스에 할당된 대역
 N : VC의 개수

새로 생성되는 토큰의 수는 <그림 1>과 같이 간격이 T_i 인 시간 슬롯의 개수와 같다. 그러나 셀의 도착시간과 T_i 슬롯 구간의 오차가 발생하므로 시간 오차를 보상하기 위해 토큰이 갱신된 시간 LT_i 는 식 (6)과 같이 계산한다.

$$LT_i(k) = Token_i \times T_i + LT_i(k-1) \quad (6)$$

VC의 셀 카운터 C_i 는 식 (7)과 같이 계산한다.

$$C_i = \min(Token_i + C_i, BL_i) \quad (7)$$

여기서, BL_i 는 Leaky Bucket의 최대 토큰 용량으로 VC의 MBS의 배수를 설정하며, VC의 셀 카운터 C_i 의 최대치이다.

2. 제안된 패킷 폐기 정책

제안된 패킷 폐기 알고리즘은 Double-EPD와 같은 두 개의 전역 임계치 LBO와 HBO를 사용한다. VC에 새로운 패킷의 첫 번째 셀이 도착 할 때 마다 VC의 셀 카운터 C_i 를 갱신한다. 셀 카운터 C_i 는 i -번째 VC가 버퍼에 받아들일 수 있는 셀 수를 의미하므로 셀 카운터에 따라 패킷의 폐기여부를 결정하여 대역 보장과 공평성을 향상시킬 수 있다. <그림 3>과 같이 버퍼의 전체 셀 점유율 QT 가 LBO이하이면 도착하는 모든 패킷은 버퍼

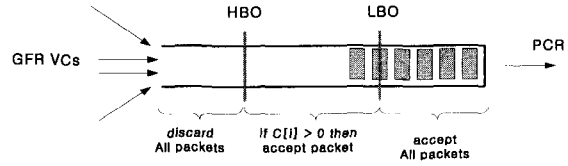


그림 2. 버퍼에 패킷의 수용 과정
Fig. 2. Packet acceptance procedure of proposed algorithm.

```

when first cell of a packet arrived:
    Token = evaluate (LT[i], T[i]); // generate new tokens by equation 3
    C[i] = min(BL[i], C[i] + Token); // update cell counter
    LT[i] = LT[i] + Token/T[i]; // update last token generation time

    if (QT > LBO && C[i] < 0) {
        drop cell;
        PS[i] = 1; // drop subsequence cell
    }
    else if (QT > HBO) {
        {
            drop cell;
            PS[i] = 1;
        }
    }
    else {
        accept cell;
        C[i]--; // decrease cell counter by 1
    }

when middle or last cell of a packet arrived:
    if (PS[i] == 1)
        drop cell;
    else {
        accept_cell;
        C[i]--; // decrease cell counter by 1
    }
    
```

그림 3. 제안된 버퍼 관리 알고리즘의 의사코드
Fig. 3. Pseudo code of proposed buffer management algorithm.

로 수용된다. 셀이 버퍼에 일단 수용되면 해당 VC의 $PS[i]$ 를 0으로 설정하고, 셀이 버퍼에 수용될 때마다 셀 카운터 C_i 를 1씩 감소한다. 만약 QT 가 첫 번째 임계치인 LBO를 초과하고 HBO 미만이고, VC의 셀 카운터가 0 이하이면 해당 VC가 셀 카운터를 모두 소비하여 할당된 대역을 초과하여 서비스 받았으므로 패킷을 폐기한다. 셀이 폐기되면 해당 VC의 $PS[i]$ 는 1로 설정하여 폐기된 셀이 속한 패킷의 모든 셀을 폐기하도록 한다. QT 가 HBO를 초과하면 심각한 폭주상황이므로 새로 도착하는 모든 패킷을 셀 카운터에 관계없이 폐기한다. <그림 2>는 제안된 버퍼 관리 알고리즘에서 패킷이 도

착할 때의 처리과정을 보여준다.

여기서 제안된 알고리즘에서 사용된 변수의 정의는 다음과 같다.

LT[i] : i-번째 VC의 가장 최근에 갱신된 시간

T[i] : i-번째 VC에 할당된 대역에 따른 셀 도착 간격

Token : 새로운 패킷의 첫 번째 셀이 도착했을 때 LT[i]와 현재 시간구간에서 발생된 토큰의 수

QT : 버퍼에 대기하고 있는 총 셀 수

PS[i] : i-번째 VC의 패킷 상태 플래그(1이면 패킷에 속한 모든 셀을 폐기하고 0이면 버퍼에 수용한다)

CF[i] : i-번째 VC의 셀 카운터

BL[i] : i-번째 VC의 Leaky-bucket 용량

IV. 시뮬레이션 및 성능평가

제안된 버퍼 관리 알고리즘의 성능평가를 위해 GFR 시뮬레이션 프로그램을 개발하였다. 개발된 GFR 시뮬레이션 프로그램은 이벤트 기반의 시뮬레이션으로 이벤트를 주고 받아 수행되는 링크, 큐, 스위치 등의 컴포넌트들로 구성이 된다. 시뮬레이션에 사용된 네트워크는 일반적으로 GFR 성능평가에 사용되는 모델로서 IP 라우터가 ATM 스위치와 직접 연결된 peer-to-peer 구성으로, 두 대의 ATM 스위치에 5대의 라우터가 결합된 네트워크로 <그림 4>와 같다. 각 라우터는 5개의 TCP가 연결되고 각 TCP는 greedy 소스로 연결 기간동안 무한 크기의 파일을 전송한다.

TCP의 MSS는 기본 960 byte로 20개의 ATM 셀을 이루며 TCP의 시뮬레이션 파라미터는 <표 1>과 같다. 송신원과 수신원은 CLP = 0인 패킷만 전송한다. 전송된 패킷은 UNI(User Network Interface)의 F-GCRA 테스트를 통해 태깅이 수행된다. 네트워크의 모든 링크는 SONET OC-1인 51.84 Mbps이며 ATM 스위치간의 지연은 5 msec이고, 라우터와 ATM 스위치간의 지연은 0.5 msec이다. <그림 4>의 왼쪽에 위치한 라우터들의 TCP는 송신원이고, 오른쪽에 대응하는 라우터의 TCP는 수신원이다. 시뮬레이션에 사용된 TCP는 New Reno 버전으로, Slow-start, Congestion avoidance와 Fast-recovery, Fast-retransmit 알고리즘^[9]이 포함된다. ATM 스위치의 버퍼용량은 4K 셀, HBO는 3K 셀이고 LBO는 1.5K 셀로 설정하여 시뮬레이션을 수행하였다. 제안한 알고리즘의 파라미터 BL_i 는 $8 \times MBS_i$ 로 설정했다.

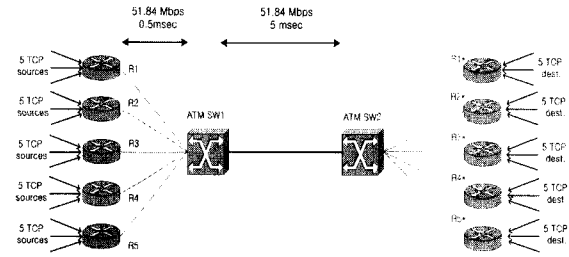


그림 4. 시뮬레이션 모델 I

Fig 4. Simulation model I.

표 1. TCP 파라미터

Table 1. TCP parameters

Parameters	Value
Window Size	65,535 bytes
Retransmission Timer	50 msec
Maximum Segment Size	960 bytes
File size	infinite(∞)

GFR에서 TCP/IP 트래픽 서비스에 대한 중요한 성능평가의 기준은 VC 수율과 공평성 지표(fairness index)이다. 공평성 지표는 각 VC들 사이에 잉여 대역을 공평하게 사용하는지를 나타내는 척도로 공평성 측정을 위해 식 (8)과 같이 공평성 지표를 계산한다^[10].

$$fairness\ index = \frac{\left(\sum_{i=1}^N x_i / f_i\right)^2}{N \times \sum_{i=1}^N (x_i / f_i)^2} \quad (8)$$

여기서 x_i : VC_i의 수율, f_i : VC_i의 공평 할당량
N : VC의 개수

여기서 f_i 는 식 (9)와 같이 MCR + equal share 방식으로 계산하였다^[1].

$$f_i = MCR_i + \frac{C_{GFR} - \sum_{j=1}^N MCR_j}{N} \quad (9)$$

여기서 C_{GFR} 은 GFR 서비스를 위해 할당된 대역이다.

1. 전체 MCR 할당에 따른 성능 평가

전체 MCR의 할당량이 성능에 미치는 영향을 알아보기 위해 전체 MCR을 20 Mbps, 30 Mbps, 40 Mbps로 할당하고 시뮬레이션을 수행하였다. 전체 MCR을 20

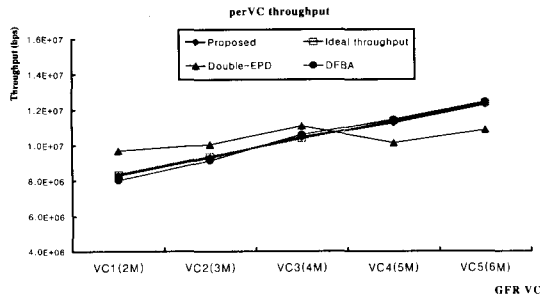


그림 5. 각 VC별 수율(전체 MCR = 20 Mbps)
Fig. 5. perVC throughput(total MCR = 20 Mbps).

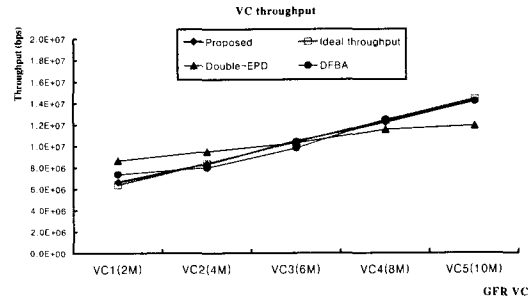


그림 6. 각 VC 수율(전체 MCR = 30 Mbps)
Fig. 6. perVC throughput(Total MCR = 30 Mbps).

표 2. 성능비교(전체 MCR = 20 Mbps)
Table 2. Performance comparison(Total MCR = 20 Mbps).

	Double-EPD	DFBA	Proposed
Total throughput	51.63 Mbps	51.65 Mbps	51.80 Mbps
Fairness Index	0.9882	0.9907	0.9994

표 3. 성능비교(전체 MCR = 30 Mbps)
Table 3. Performance comparison(Total MCR = 30 Mbps).

	Double-EPD	DFBA	Proposed
Total throughput	51.74 Mbps	51.66 Mbps	51.74 Mbps
Fairness Index	0.9702	0.9942	0.9995

Mbps로 설정했을 때 각 VC의 MCR은 2, 3, 4, 5, 6 Mbps이다. <그림 5>와 <표 2>는 전체 MCR을 20 Mbps로 설정했을 때의 성능을 보여준다. <그림 5>에서 DFBA와 제안된 알고리즘의 경우 각 VC의 수율이 이상적인 수율과 근접하여 좋은 결과를 나타냈음을 알 수 있다. 그러나 Double-EPD의 경우, 낮은 MCR을 할당 받은 VC나 높은 MCR을 가진 VC의 경우 비슷한 수율을 보여 공정성이 나쁨을 알 수 있다. <표 2>는 알고리즘 별 공정성 지표와 네트워크 수율을 보여준다. 세 가지 알고리즘 중 Double-EPD가 0.9882로 가장 나쁘게 나오고, DFBA와 제안된 알고리즘의 경우 0.99이상의 좋은 성능을 나타냈다. Double-EPD가 MCR에 따른 대역할당이 나쁨에도 불구하고 0.9882의 비교적 높은 공정성 지표를 나타낸 이유는 전체 MCR의 할당이 20 Mbps로 네트워크 용량이 비해 낮고 각 VC에 할당된 MCR의 차이가 1Mbps로 작기 때문에 공정성 지표에서 성능 저하가 작았다.

<그림 6>과 <표 3>은 전체 MCR을 30 Mbps로 설정하고 각 VC의 MCR을 2, 4, 6, 8, 10 Mbps로 설정했을 때 VC 별 수율을 보여준다. Double-EPD의 경우 낮은 MCR의 VC가 상대적으로 높은 수율을 나타냈고 높은 MCR의 경우 할당된 대역보다 낮은 수율을 나타내어 낮은 성능을 보인 반면 DFBA와 제안된 알고리즘의 경

우 이상적인 수율 그래프에 근접하여 0.99 이상의 좋은 공정성 성능을 보였다. <그림 7>과 <표 4>는 전체 MCR을 40 Mbps로 설정하고 각 VC의 MCR을 2, 5, 8, 11, 14 Mbps로 설정했을 때의 결과를 보여준다. Double-EPD와 DFBA의 경우 낮은 MCR의 VC가 상대적으로 높은 수율을 나타냈고 높은 MCR의 경우 할당된 대역보다 낮은 수율을 나타내어 공정성이 나빠졌으나 제안된 알고리즘의 경우 이상적인 수율에 근접하여 높은 공정성을 보여준다. 특히 Double-EPD와 DFBA의 경우 MCR이 14Mbps인 VC 5의 경우 수율이 MCR이하로 서비스 받아 GFR이 요구하는 최소전송률 보장에 실패했으나 제안된 알고리즘의 경우 모든 VC의 MCR을 보장할 수 있었다. <표 4>에서 Double-EPD와 DFBA는 전체 MCR이 20, 30Mbps일 때보다 공정성 지표가 현저하게 낮아졌음을 알 수 있다. MCR의 차이와 전체 MCR 할당이 클수록 제안된 알고리즘은 우수한 성능을 보인 반면, DFBA는 VC에 할당된 MCR의 차이가 클수록 공평한 서비스를 하지 못했으며, Double-EPD는 모든 경우에서 가장 나쁜 성능을 보였다.

2. 지연시간이 성능에 미치는 영향

TCP는 망의 폭주상황에 유연하게 대처하기 위해 폭주 회피와 느린 시작, 두 가지 흐름제어 기법을 사용한다. 이 두 가지 기법은 패킷의 손실이 발생했을 경우 원

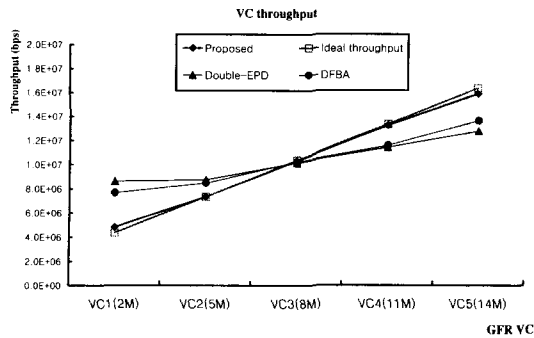


그림 7. VC 수율 (전체 MCR = 40 Mbps)
Fig. 7. perVC throughput(Total MCR = 40 Mbps).

표 4. 성능비교(전체 MCR = 40 Mbps)
Table 4. Performance comparison(Total MCR = 40 Mbps).

	Double-EPD	DFBA	Proposed
Total throughput	51.73 Mbps	51.63 Mbps	51.67 Mbps
Fairness Index	0.8770	0.9164	0.9977

도구 크기를 줄여서 전송 속도를 낮춤으로써 망의 폭주 상황을 회피하는 기법으로 종단 간의 전송 지연 시간에 밀접한 관계가 있다^[8]. 두 가지 알고리즘이 수행이 되면 전송속도를 줄이고, 다시 시간이 지남에 따라 속도를 증가시키는 방법으로 RTT가 짧은 연결일수록 전송속도의 증가시간이 짧아져서 상대적으로 RTT가 긴 연결보다 서비스를 많이 받을 수 있다.

GFR VC의 지연시간이 서로 다를 때 TCP 성능에 미치는 영향을 알아보기 위해 지연시간을 두 종류로 나누어 시뮬레이션을 수행하였다. <그림 8>과 같이 시뮬레이션 네트워크를 구성하였다. 여섯 대의 라우터를 지연시간이 서로 다른 두 그룹으로 구성해서 첫 번째 그룹은 GFR VC1 ~ VC3으로 지연시간은 11 msec, 두 번째 그룹은 GFR VC4 ~ VC6으로 지연시간을 21 msec로 설정하였다. 각 그룹에서 MCR은 각각 4, 6, 8 Mbps로 설정하였다.

<그림 9>는 지연시간이 11 msec와 21 msec로 설정했을 때 Double-EPD, DFBA와 제안된 알고리즘의 시뮬레이션 결과이다. 점선으로 표시된 그래프는 지연시간이 11 msec인 GFR VC의 VC 수율이고 실선으로 표시된 그래프는 지연시간이 21 msec인 GFR VC의 수율이다. Double-EPD와 DFBA의 경우에 같은 MCR을 가진

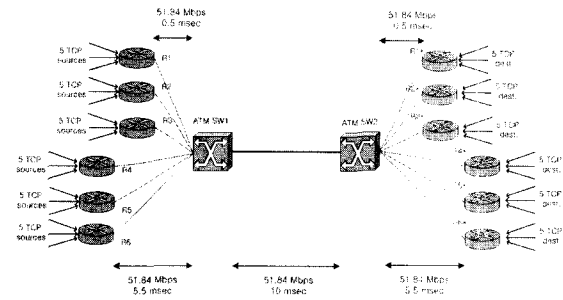


그림 8. 시뮬레이션 모델 II
Fig. 8. Simulation model II.

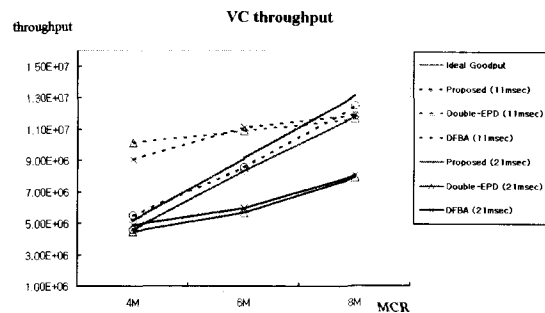


그림 9. 지연환경에서 VC 수율
Fig. 9. perVC throughput with different delay config.

표 5. 성능비교(지연환경)
Table 5. Performance comparison(different delay config.)

	Double-EPD	DFBA	Proposed
Total throughput	50.84 Mbps	51.08 Mbps	51.33 Mbps
Fairness Index	0.8283	0.8721	0.9960

GFR VC를 비교했을 때 지연시간이 짧은 첫 번째 그룹의 VC 수율이 지연시간이 큰 두 번째 그룹보다 월등히 높았다. 특히 지연시간이 21 msec이고 MCR이 6 Mbps로 설정된 VC의 경우 수율이 5 Mbps 정도로 MCR 미만으로 서비스 받았다. 반면에 제안된 알고리즘의 경우에 지연시간에 관계없이 이상적인 수율에 근접해서 GFR VC가 요구하는 MCR과 공평성을 제공하였다.

<표 5>는 전체 네트워크 수율과 공평성을 수치로 표현한 것으로 Double-EPD와 DFBA의 공평성이 각각 0.8283, 0.8721로 지연시간이 같은 경우에 비해 공평성이 현저히 낮아졌다. 반면 제안된 알고리즘은 0.9960으로 높은 공평성 지표를 나타내어 지연시간이 TCP에 미치지

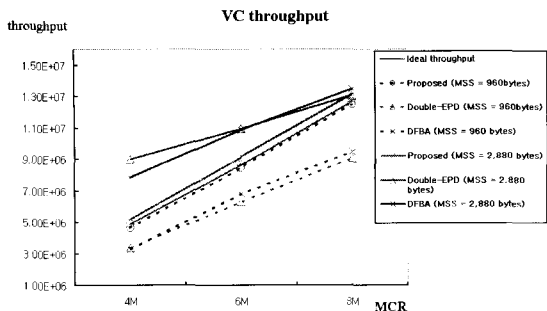


그림 10. MSS에 따른 VC 수율
Fig. 10. perVC throughput with different MSS config.

표 6. MSS에 따른 성능비교
Table 6. Performance comparison with different MSS config.

	Double-EPD	DFBA	Proposed
Total throughput	51.77 Mbps	51.77 Mbps	51.77 Mbps
Fairness Index	0.8552	0.8995	0.9994

는 영향을 최소화하여 Double-EPD와 DFBA보다 전체 수율과 공평성에서 우수한 성능을 보여준다.

3. MSS가 성능에 미치는 영향

TCP의 최대 세그먼트 크기인 MSS가 성능에 미치는 영향을 알아보기 위하여 VC를 두 그룹으로 나누어 각 그룹 당 서로 다른 MSS를 설정하였다. 첫 번째 그룹은 GFR VC1 ~ VC3으로 MSS가 960 byte, 두 번째 그룹은 GFR VC4 ~ VC6으로 MSS를 2,880 byte로 첫 번째 그룹보다 3배 크게 설정하였다. 각 그룹에서 MCR은 각각 4, 6, 8 Mbps로 설정하였다. 각 종 단간의 지연시간은 6 msec로 모든 VC가 동일하다.

<그림 10>은 MSS에 따른 Double-EPD, DFBA와 제안된 알고리즘의 시뮬레이션 결과이다. 점선으로 표시된 그래프는 MSS가 960 byte인 GFR VC의 VC 수율이고 실선으로 표시된 그래프는 MSS가 2,880 byte인 GFR VC의 수율이다.

Double-EPD와 DFBA 알고리즘의 경우, MSS가 2,880 byte로 설정된 두 번째 그룹의 VC들이 960 byte로 설정된 첫 번째 그룹의 경우 보다 VC의 수율이 높게 나왔다. 이런 결과가 나온 중요한 원인은 종단의 TCP 계층에서 윈도우 크기를 증가할 때 세그먼트 단위

로 증가하기 때문이다. 즉, 세그먼트 크기가 큰 연결이 세그먼트 크기가 작은 연결보다 빠르게 윈도우가 증가하기 때문에 서비스를 많이 받게 된다. 기존 알고리즘은 패킷을 버퍼에 수용할 때 패킷의 크기를 고려하지 않기 때문인데 제안된 알고리즘의 경우 셀 카운터 변수에 따라서 버퍼에 수용되기 때문에 MSS에 영향을 받지 않고 공평한 서비스가 이루어졌다. MSS가 960 byte이고 MCR이 4Mbps로 설정된 VC의 경우 Double-EPD와 DFBA에서 MCR 이하로 서비스 받아 GFR이 요구하는 MCR 보장에 실패했음을 알 수 있다.

<표 6>은 공평성 지표와 전체 네트워크 수율을 보여준다. Double-EPD와 DFBA의 공평성이 각각 0.8552, 0.8995로 MSS가 같은 경우에 비해 공평성이 현저히 낮아졌다. 반면 제안된 알고리즘은 0.9994으로 높은 공평성 지표를 나타내어 지연시간이 TCP에 미치는 영향을 최소화하여 Double-EPD와 DFBA보다 공평성에서 우수한 성능을 보여준다.

V. 결 론

B-ISDN에서 Web, 대용량 파일 전송, 실시간 스트리밍 서비스 등과 같은 인터넷 데이터의 폭발적인 증가로 인해 ATM 네트워크에서 인터넷과 같은 "best-effort" 트래픽을 효율적으로 전송해야 한다. 그 결과로 ATM Forum에서 인터넷 트래픽과 같은 "best-effort" 트래픽을 서비스하기 위해 새로운 GFR 클래스를 제안했다. 본 논문에서는 ATM 네트워크에서 GFR을 통해 TCP/IP 트래픽 서비스를 위한 새로운 버퍼 관리 알고리즘을 제안하였다. 제안된 알고리즘은 Leaky-Bucket 알고리즘을 기반으로 기존의 Leaky-Bucket의 복잡성과 프로세싱 부하를 줄이면서 GFR VC가 요구하는 MCR을 보장하고 높은 공평성을 제공한다. 컴퓨터 시뮬레이션을 통하여 기존의 Double-EPD, DFBA와 성능을 비교하여 공평성과 전체 수율 면에서 우수한 성능을 보여주었다.

참 고 문 헌

[1] The ATM Forum Technical Committee, Draft TM 4.1 Traffic Management Specification, ATM Forum/BTD-TM-02.02, Dec. 1998.
[2] R. Goyal, R. Jain, S. Fahmy and B. Vandalore, "Simulation Experiments with Guaranteed Frame

- Rate for TCP/IP traffic," ATM Forum/97-0607.
- [3] R. Guerin and J. Heinanen, "UBR+ Service Category Definition," ATM Forum, ATM 96-1598, Dec. 1996.
- [4] R. Guerin and J. Heinanen, "UBR+ Enhancement," ATM Forum, ATM97-0015, Feb. 1997.
- [5] R. Goyal, R. Jain, S. Kalyanaraman, et. "TCP Selective Acknowledgement and UBR Drop Policies to Improve ATM-UBR Performance over Terrestrial and Satellite Networks," Proc. ICC'97, June 1997.
- [6] J. Heinanen and K. Kilikiki, "A Fair Buffer Allocation Scheme", Computer Communication 21, pp 220~226, 1998.
- [7] O. Bonaventure. "Providing Bandwidth Guarantees to Internetwork Traffic in ATM Networks," Proc. of IEEE ATM98 workshop, May 1998.
- [8] W.Stevens. "TCP slow start, congestion avoidance, fast retransmit and fast recovery algorithms", Internet RFC 2001, Jan. 1997.
- [9] V. Jacobson, "Congestion avoidance and control," In Proc. ACM SIGCOM88, pp 314~329, Aug. 1988.
- [10] Rohit Goyal, Raj Jain, et. al "Improving the performance of TCP over the ATM-UBR service", Journal of Computer Communications Vol. 21, No. 10, July 1998.

저 자 소 개

金 觀 雄(正會員) 第38卷 SD編 第4號 參照

田 炳 實(正會員) 第38卷 SD編 第4號 參照

金 辯 坤(正會員) 第39卷 TC編 第6號 參照