

ATM상의 TCP 패킷 폐기정책에 따른 공정성 개선에 관한 연구

육 동 철[†] · 박 승 섭^{††}

요 약

최근 초고속 인터넷 상에서 여러 가지 응용프로그램과 서비스들의 증가로, ATM 백본 네트워크는 주된 해결책이 되어왔다. TCP는 현재 인터넷의 상위 어플리케이션을 지원하는 표준 프로토콜이고, 트랜스포트 계층에서 윈도우 기반 프로토콜 흐름제어를 사용하고 있다. TCP의 데이터가 ATM상의 UBR서비스를 사용할 경우, ATM 스위치 버퍼관리에 의해서 제어되며, 하나의 셀 손실은 전체 패킷의 손실을 가져온다. 이러한 현상은 대부분 TCP 성능 저하를 가져오고, 충분한 QoS를 제공하지 못한다. 이런 문제를 해결하기 위해 Tail Drop, EPD, PPD, SPD, FBA와 같은 셀 폐기 정책이 ATM 상의 TCP 성능을 개선하기 위해서 제안되어져 왔다.

본 논문에서는 종단간의 TCP 공정성 개선을 위해서, 두 개의 고정된 임계치를 사용한 폐기 정책 알고리즘을 제안하고, 동일한 시뮬레이션 환경 하에서 기존의 알고리즘들과 비교하였다. VC의 수가 증가하더라도, 제안한 알고리즘이 각 VC에 대해 다른 알고리즘과 비교해서 대역폭 할당이 보다 공정하게 이루어짐을 시뮬레이션 결과로 알 수 있었다.

A Study on Improving the Fairness by Dropping Scheme of TCP over ATM

Dong-Cheol Yuk[†] · Seung-Seob Park^{††}

ABSTRACT

Recently, the growth of applications and services over high-speed Internet increase, ATM networks as wide area back-bone has been a major solution. The conventional TCP suite is still the standard protocol used to support upper application on current Internet and uses a window based protocol for flow control in the transport layer. When TCP data uses the UBR service in ATM layer, the control method is also buffer management. If a cell is discarded in ATM layer, one whole packet of TCP will be lost. Which is responsible for most TCP performance degradation and do not offer sufficiently QoS. To solve this problem, Several dropping strategies, such as Tail Drop, EPD, PPD, SPD, FBA, have been proposed to improve the TCP performance over ATM.

In this paper, to improve the TCP fairness of end to end, we propose a packet dropping scheme algorithm using two fixed threshold. Under similar condition, we compared our proposed scheme with other dropping strategies. Although the number of VC is increased, simulation results showed that the proposed scheme can allocate more fairly each VC than other schemes.

1. 서 론

ATM 포럼에서는 ATM서비스 범주를 음성, 동영상,

데이터에 따라 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 서비스 범주로 표준화 하였다[1]. 그러므로 TCP 데이터와 같이 손실에 민감한 트래픽은 ABR과 UBR상에서 안정된 품질로 전송

† 준 회 원 : 부경대학교 대학원 전자계산학과
†† 정 회 원 : 국립부경대학교 전자컴퓨터정보통신공학부 교수
논문접수 : 2000년 11월 3일, 심사완료 : 2000년 12월 29일

되어야 한다[2].

그러나 TCP는 현재 인터넷의 상위 어플리케이션을 지원하는 프로토콜이지만, ATM구조에서의 효율적인 동작과 어떻게 QoS를 제어하는가는 ABR과 UBR 상에서의 TCP 평가로 분석되어 왔으며, 여전히 연구 과제로 남아있다[3, 4].

특히 TCP의 하위 계층인 ATM에서 하나의 셀 데이터 손실은 상위 계층에서 심각한 문제를 발생시킬 수 있다. 그 이유는 서로 두 계층간의 프로토콜(Protocol) 형태 면에서 가변 길이 데이터와 고정길이 데이터 포맷 차이와 TCP에서 사용되는 제어방식과 ATM에서 수행되고 있는 제어방식은 각기 그 서비스 범주에 맞게 서로 다른 방식으로 수행되고 있다. 그러므로 ATM층에서 한 개의 셀 손실은 상위 TCP층에서 하나의 패킷에 손실을 가져온다.

Raj Jain의 plain UBR 상에서의 TCP 관련한 시뮬레이션 연구 결과에 따르면, zero loss를 위해서는 UBR 버퍼는 모든 TCP 소스의 윈도우 크기의 합만큼 필요하며, 요구되는 UBR 버퍼 크기가 소스의 수에 의존적임을 나타내었다. 그리고 높은 성능을 위해, TCP 소스는 다른 TCP 소스가 낮은 사용율을 가지고 있을 동안에 링크 이용률을 최대한 이용할 수 있어야 한다고 보고하였다[5, 6].

Romanow와 S. Floyd는 버퍼에서 효율적인 처리를 위해 EPD(Early Packet Discard), PPD(Partial Packet Discard)를 제안하였다. EPD나 PPD는 UBR상에서 TCP 처리율을 매우 향상시키게 되었으나, 공정성 면에서는 데이터 전송을 하는 컨넥션 간 대역폭 할당에 불균형이 발생되었다[7-9].

Rohit Goyal은 효율적인 처리율과 공정성 향상을 위해 per-VC 당 패킷 폐기 정책인 SPD>Selective Packet Discard)를 제시하였다. 그리고 TCP 소스 5~15개를 가진 망 구조에서 plain UBR, EPD, SPD, FBA와 FRR(Fast Retransmit and Recovery)을 사용하여 비교 분석하였고[10], 그 외 성능 향상을 위한 per-VC 당 버퍼 할당 방식인 FBA(Fair Buffer Allocation)가 제안되었으며, 여러 연구와 비교 분석되었다[8, 11].

이러한 여러 방식을 기본으로 하여 본 논문은 각 컨넥션에 대해 공정성을 1로 기준 할 때, 위의 폐기 정책보다 나은 방식을 제안하고, EPD, SPD, FBA를 같은 네트워크 시뮬레이션 망 구조 조건에서, VC 수를 최대 5~70까지를 사용하고, TCP의 컨넥션과 스위치

간의 지연과 스위치와 스위치간의 지연, MTU 크기에 따른 성능을 비교 분석하였다.

본 논문 구성은 서론에 이어, 2장에서는 TCP 흐름 제어와 폭주제어 방식, EPD, SPD, FBA 알고리즘에 관련한 설명과 3장에서는 공정성 개선을 위해 제안 방식의 새로운 파라미터와 처리 방법에 관한 내용과, 4장과 5장에서는 시뮬레이션 환경 설정과 결과분석을 나타내었고, 마지막으로 6장에서는 결론에 대해 서술하였다.

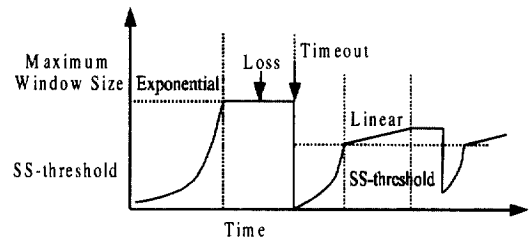
2. 관련연구

본 장에서는 TCP의 제어 방식과 ATM의 셀 폐기 방식들에 대하여 서술한다.

2.1 Slow Start 방식

Slow Start는 폭주 윈도우를 사용한다. 새로운 연결이 설정되었을 때 TCP 컨넥션은 폭주 윈도우를 세그먼트크기와 같은 1로 초기화하고, 수신지에서 ACK가 수신될 때마다 폭주 윈도우를 최대값까지 1씩 증가시킨다. 다시 말해서 세그먼트를 두 배씩 증가시킨다. 따라서 ACK값이 지수 증가하기 때문에 실제로 폭주 윈도우도 지수 증가한다[15].

(그림 1)에서 손실이나 시간초과가 발생하면 SS-threshold(Slow Start threshold)를 현재 폭주 윈도우 크기의 반으로 설정하고 폭주 윈도우의 크기는 1로 설정하며, 다시 Slow start 과정을 폭주 윈도우 크기가 SS-threshold까지 증가시키며, 그 다음 단계로는 수신된 ACK마다 폭주 윈도우를 한 개씩 증가시킨다.



(그림 1) Slow start와 congestion avoidance 단계의 폭주 윈도우의 크기 변화

2.2 Fast retransmission과 Fast recovery 방식

Fast retransmit은 TCP 버전(예 : Reno, New Reno, Tahoe, Vegas)에 따라라도 약간의 차이는 있지만 손

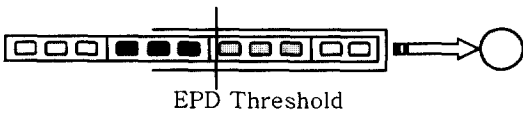
실린 세그먼트를 빨리 복구한다는 의미에서는 차이가 없다. 송신측은 순차 번호 순서대로 세그먼트를 송신하며, 수신측에서는 순차 번호 순서대로 세그먼트를 수신한다.

만약 수신측에서 순차 번호 순서에 어긋난 세그먼트가 수신되면, 수신측은 수신되지 않은 순차 번호로 ACK를 송신하며, 이는 올바른 순차번호의 세그먼트가 수신 될 때까지 반복된다. 따라서 송신측에서 중복된 ACK가 3개 이상 수신되면 이는 세그먼트의 손실로 간주되어, 즉시 손실된 세그먼트를 재전송하고 타이머가 만료될 때까지 기다려야하는 시간을 줄이는 방법이다.

Fast recovery는 세그먼트의 손실 시 폭주 윈도우의 1/2을 SS-threshold로 설정하고, 폭주 윈도우의 크기를 1로 설정하고서 slow start 단계로 들어가는 방법이 보수적이라고 판단하여, 이러한 방법 대신 세그먼트의 손실 시 폭주 윈도우 크기의 1/2를 현재 폭주윈도우의 크기로 설정하고, 곧바로 폭주 회피 단계로 들어가고 Slow start 단계를 제거하는 방법이다.

2.3 EPD 알고리즘

Romanow와 S Floyd는 ATM 망에서 TCP 서비스를 사용해서 패킷 기반의 TCP의 성능을 분석하였다. 그들의 성능 분석 결과, 폭주 발생으로 인한 셀 폐기는 성능이 저하될 수 있음을 보였다. 이러한 낮은 처리율은 전송로 상에 파괴된 패킷 전송으로 말미암아 대역폭 낭비를 가져오기 때문이다. 이 문제를 해결하기 위해 그들이 제안한 EPD는 ATM 망에서 TCP의 성능을 개선 시켰다.



(그림 2) EPD 방식과 EPD 임계치

(그림 2)와 같이 EPD 구현은 스위치의 버퍼에 임계치를 설정하는 것이다. 스위치는 큐가 주어진 임계치를 초과했을 때 들어오는 패킷들을 폐기한다. 여기서 임의로 패킷을 폐기하는 것이 아니라 첫 번째 셀이 있으면 같은 가상 채널 식별자를 가진 셀을 EOM 까지 전체 폐기한다. 만약 전체 버퍼가 꽉 찬 상태가 아닐지라도, 첫 번째 셀이 폐기되지 않으며, 나머지 셀들도 폐기되지 않는다. 그리고 EPD는 PPD와 많이 연결되

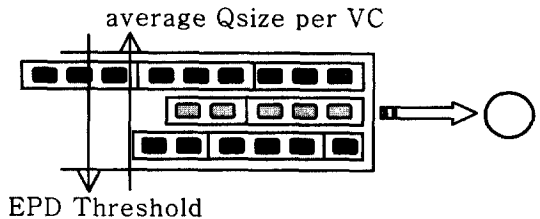
어서 사용되어진다. 참고 논문[7]에서는 EPD는 PPD와 Tail Drop보다 향상된 성능을 보였다. 그러나 EPD는 각 TCP 컨넥션간의 각 셀의 VCI에 대한 정보를 사용하지 않는 알고리즘이기 때문에 공정성에 대한 문제가 있다.

2.4 SPD 알고리즘

SPD는 EPD의 공정성을 개선하기 제안되었다. 여러 개의 TCP 컨넥션중에서 하나의 TCP 컨넥션이 패킷 전송을 한 후에, 수신측 ACK에 따라 Slow start 단계를 수행하기 위해 윈도우 크기를 증가시킨다. 그러나 다른 TCP 컨넥션도 동시에 전송할 경우 버퍼 오버플로우에 의해 셀이 폐기되는 현상을 발생시킨다. 또는 다른 TCP 컨넥션이 타임아웃에 의해 폭주윈도우를 줄여야 하는 현상이 야기된다. 이와 같은 현상에 의해 TCP 컨넥션은 대역폭에 있어 불공평한 할당을 받는다.

이 문제를 줄이기 위해, SPD는 버퍼에 있는 각 VC의 셀 수와 활성화된 VC의 수에 대한 정보를 이용하였다. 여기서 버퍼에 적어도 하나 이상의 셀을 가지다면, 그 VC는 활성화된 상태이고 VC의 수는 TCP 컨넥션의 수이다. 공정한 할당은 활성화된 VC의 수에 현재 버퍼의 큐 길이를 바탕으로 계산된다.

(그림 3)에서 각 VC마다 스위치에 들어온 TCP 패킷을 VCj라 가정하고, 만약 VCj에 대하여 현재 버퍼 크기가 $Q \geq Th$ 이거나 $Q_j \geq [Th]$ 인 경우 첫 번째 셀은 폐기된다. 그리고 나머지 처리는 EPD와 같다. 여기서 그림은 하나의 버퍼를 각 VC의 셀들로 세분화해서 묘사되었으며, Qj는 스위치 버퍼의 VCj에 해당하는 셀 수이며, [Th]는 $K * Q/N$ 으로 정의할 수 있고, 이는 VC 당 평균 큐 길이가 된다.



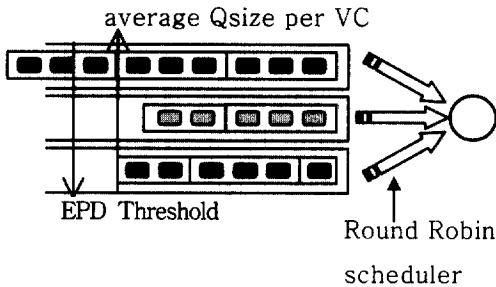
(그림 3) SPD 방식

N은 활성화된 VC의 수이고, K는 제어 파라미터인데 전형적으로 $1 \leq K \leq 2$ 로 설정한다. 본 연구에서는 K를 일반적으로 사용하는 값 1로 설정하였다.

2.5 FBA 알고리즘

스위치에 최대 연결된 VC의 수만큼 버퍼를 사용해서 활성화된 VC의 수에 현재 버퍼의 큐길이를 가지고 계산된다. 각 Q_j 는 VCj의 큐 길이로 대응되며, 전체 큐 길이 Q 가 $Q \geq Th$ 이고 $Q_j \geq [Th]$ 일 경우, VCj의 새로운 패킷은 폐기된다. $[Th]$ 는 VC당 평균 큐 길이가 되며, $K \times Q/N$ 으로 정의한다. N 은 활성화된 VC의 수와 K 는 제어 파라미터 의미한다. K 값은 SPD와 같이 1로 설정하였다.

FBA에서 모든 VC의 큐는 라운드 로빈 스케줄에 의해 처리된다. 이때 각 라운드 로빈에 의해 셀이 처리될 경우, 오직 하나의 VC 큐 셀이 처리된다. (그림 4)와 같이 각 VC는 스위치에서 분리된 버퍼마다 처리가 된다.



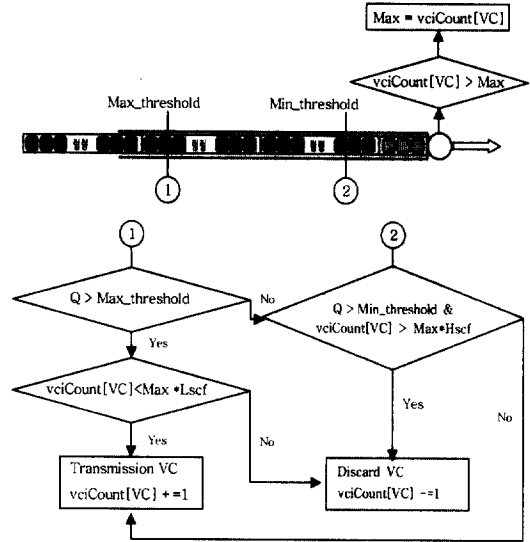
(그림 4) FBA 방식

3. 공정성 개선을 위한 제안한 알고리즘

앞장에서 설명한 알고리즘을 기반으로 본 연구는 (그림 5)와 (그림 6)과 같이 제안한 알고리즘의 순서도와 의사코드를 나타낸다.

제안한 알고리즘의 중요한 내용은 EPD나 SPD와 같이 단일 FIFO 큐를 사용하고, SPD와 FBA와 같이 두 개의 고정된 임계치를 사용한다. 그러나 SPD와 FBA는 두 개의 임계치 중 하나의 임계치는 각 활성화된 VC의 수와 제어 파라미터에 의해 유동적으로 변한다.

제안한 알고리즘은 들어오는 셀에 대하여 고정된 두 임계치와 각 VC의 정보에 따라 선택적으로 셀을 폐기하는데, 버퍼에 셀들이 폐기 또는 승인되는 정보를 간단히 vciCount에 임시 저장함으로써 임의의 VCj의 패킷이 얼마나 많이 폐기되고, 임의의 VCj가 다른 VCj+n 보다 얼마만큼의 패킷이 전송되었는가의 척도에 의한 방식이다.



(그림 5) 제안한 알고리즘의 순서도

```

When a cell is coming to an ATM switch:
if the cell is the first cell of a packet
  if Q >= Max_threshold
    if vciCount[VC] < max/Low selective control
      factor) vciCount[VC] ++;
    accept the cell into the single FIFO queue
  else
    discard the cell
    vciCount[vc] --
  else
    if Q >= Min_threshold)
    if vciCount[VC] > max/High selective control
      factor)
    discard the cell
    vciCount[vc] --
  else
    vciCount[VC] ++;
    accept the cell into the single FIFO queue
  else
    if any cell of the packet has been discard
    discard the cell
  else
    if Q >= Qmax
    discard the cell
  else
    accept the cell into the single FIFO queue
    
```

(그림 6) 제안한 알고리즘의 의사코드

간단히 요약하면 전송량이 많은 VC가 있을 경우 폐기 확률이 높고, 전송량이 작은 VC에 더 많은 전송 기회를 주는 것이다.

버퍼에 어떤 첫 번째 셀이 들어 올 경우에 Max_threshold는 EPD의 임계치와 같은 역할을 한다. 그러나 vciCount의 정보가 만약 max/Low selective control factor 작을 경우 EPD, FBA, SPD의 첫 번째 임계치와 같이 폐기를 하지 않고 전송된다. 여기서 max/Low selective control factor는 전송량이 작은 VC를 구체해 주는 역할을 한다.

그리고, 버퍼의 큐에 Min_threshold 보다 큰 첫 번째 셀이 있을 경우, vciCount는 max/High selective control factor와 비교하고 만약 vciCount가 클 경우 폐기된다. 이는 전송량이 많은 VC로 간주하여 전송량을 평균치만큼 조종하기 위함이다.

제시된 알고리즘에는 여러 개의 파라미터가 추가되었다. 추가된 파라미터에 대한 설명은 <표 1>과 같이 나타내었다.

<표 1> 새로 추가된 파라미터에 대한 내용

새로 추가된 파라미터	내 용
Min_threshold	Max_threshold를 지난 첫 번째 셀을 다시 검색하고, vciCount[VC]의 정보를 이용해서 승인 횟수가 max ÷ Hscf 보다 클 경우 폐기함으로써 각 VC 간의 공정성을 확보하기 위한 최소 임계치
max	패킷이 버퍼 처리가 승인 될 때 vciCount[VC]의 값이 올라간다. 여기서 최고 높은 값을 가질 때, max 값 부여
High selective control factor (Hscf)	최대 max를 기준으로, max ÷ Hscf보다 큰 vciCount[VC]를 검색할 때 사용되는 제어 factor
Low selective control factor (Lscf)	최대 max를 기준으로, max ÷ Lscf보다 작은 vciCount [VC]를 검색할 때 사용되는 제어 factor

그리고 추가적으로 프로그램 내부에 정수형 vciCount [VC]변수를 선언하였다. 이 변수는 각 VC마다 컨넥션 설정이 이루어지면, 설정된 VC의 개수만큼 배열크기가 생성되고, 패킷이 승인된 개수만큼 배열의 값은 정수형으로 증가되어서 저장된다.

버퍼에 패킷이 들어 올 때마다 두 개의 임계치는 패킷의 첫 번째 셀을 계속 검색한다. 그리고 승인된 패킷이 있으면 각 vciCount[VC]에 값을 증가시킨다. 이때 스위치는 최고 많이 승인된 vciCount[VC]의 갯수를 조사하여 max값을 갱신한다.

첫 번째 Max_threshold에서 하는 역할은 기존 EPD의 역할과 유사하다. 단지 폐기를 할 때, Low se-

lective control factor 파라미터를 사용하고, Min Threshold는 승인된 패킷이 있을 지라도 High selective control factor 파라미터를 사용해서 다른 VC 보다 패킷 승인 처리가 많을 경우에는 폐기를 시킨다. 추가된 파라미터인 Min_Threshold는 Max_threshold - 1000 값만큼 설정하였고, High selective control factor 값은 1/3, Low selective control factor 값은 2/3로 각각 설정하여 실험하였다.

4. 시뮬레이션 환경

본 장은 시뮬레이션 망 모델과 파라미터에 대해 설명한다. 시뮬레이션 도구로써는 C++로 작성되어진 사건 중심(event-driven)의 Yats ATM 시뮬레이션 도구 [10]를 기반으로 여러 가지 헤더파일과 SPD와 FBA 알고리즘 오브젝트 파일을 추가하여 구성하였다.

4.1 망 모델

UBR서비스를 사용하는 TCP의 성능 분석을 위해서 하나의 병목 구간을 가지는 peer to peer 모델로 설정하였다. 전송방향은 단방향 트래픽으로 데이터를 전송하게 하였다. 트래픽 유형은 greedy 형태로 동시에 전송하게 하였으며, 두 스위치간의 링크 속도는 150Mbps로 설정하였다.

TCP 컨넥션의 수, 즉 VC 수는 가변적으로 최대 n만큼의 수를 가질 수 있으나, 본 연구에서는 n을 최소 5에서 최대 70까지의 범위로 설정하고, TCP의 컨넥션과 스위치간의 지연과 스위치와 스위치간의 지연, MTU(Maximum Transmission Unit)크기도 가변적으로 설정하였다. 시뮬레이션 시간은 총 3초 동안 실험을 하였다.

4.2 시뮬레이션 파라미터

각 UBR, TCP 컨넥션, UBR 스위치의 기본 파라미터는 아래와 같이 설정하였다.

- TCP Application
 - data frame = 8000 bytes
- TCP
 - Mean Packet Processing = 300 μs
 - Send/Receiver buffer size = 64 Kbytes
 - MTU size = 가변 (512, 1024, 4096, 8200) bytes

Default timeout = 500 msec
 Minimum RTO = 100 msec
 Maximum receiver window size = 64 Kbyte
 Fast retransmission and recovery = on

- **UBR-Switch**
 Output Buffer Size = 4096 cells
 EPD threshold = 3500 cells
- **SDP and FBA control parameter**
 K = 1
 SPD & FBA threshold = 3500 cells
- **제안 알고리즘의 파라미터**
 High selective control factor = 1/3
 Low selective control factor = 2/3
 Max threshold = 3500 cells
- **Delay**
 TCP connection & Switch =
 10, 50, 100, 200, 300 time slots
 Switch & Switch =
 100, 300, 500, 700, 1000 time slots

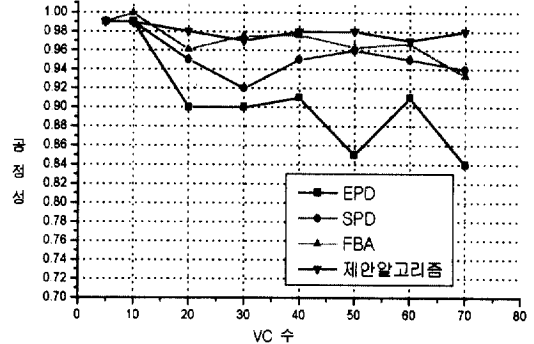
5. 결과 분석

성능 평가 요소로서는 각 VC 수에 따른 처리율과 공정성을 기준으로 하였다. 평가 요소인 공정성은 서로 VC 간 대역폭 할당에 있어 공평하게 할당받았는가를 평가하는 기준으로 다음 식과 같이 주어진다[13, 14]. 여기서, $x_i = T_i / O_i$ 로 $T_i (i = 1 \dots n)$ 는 각 VC마다 성공적으로 도착된 패킷의 측정된 처리값으로 단위시간당 전송 받은 양을 바이트로 계산하였고,

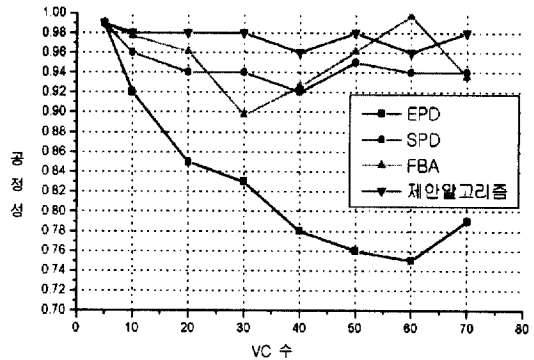
$$Fairness = \frac{(\sum x_i)^2}{n \times (\sum x_i^2)}$$

$O_i (i = 1 \dots n)$ 는 max-min의 최적의 처리율이고, n 은 전체 VC의 수이다.

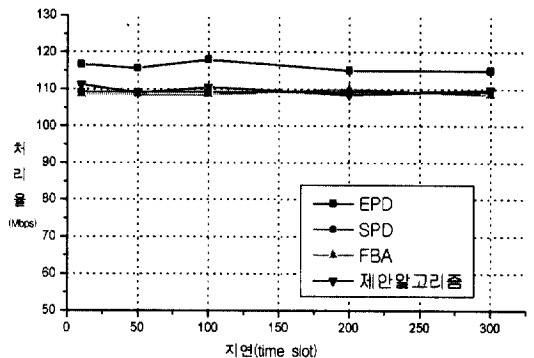
(그림 7)과 (그림 8)은 VC의 수에 따른 평가로써, VC의 수가 5~70이고, MTU 크기가 각각 512 bytes와 1024 bytes 일 경우, 스위치와 스위치 지연은 300 time slots과 TCP 컨넥션과 스위치 지연은 10 time slots으로 설정하였다. 여기서 기존 EPD의 공정성 문제를 SPD, FBA 알고리즘이 해결하였음을 알 수 있다. 그러나 본 연구에서 제안한 알고리즘은 VC 수가 증가할지라도 다른 알고리즘의 공정성 평가보다 나은 결과를 보이고 있다.



(그림 7) MTU가 512 bytes 일 경우의 공정성 비교



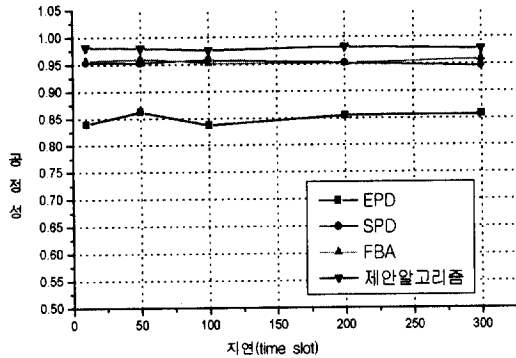
(그림 8) MTU가 1024 bytes 일 경우의 공정성 비교



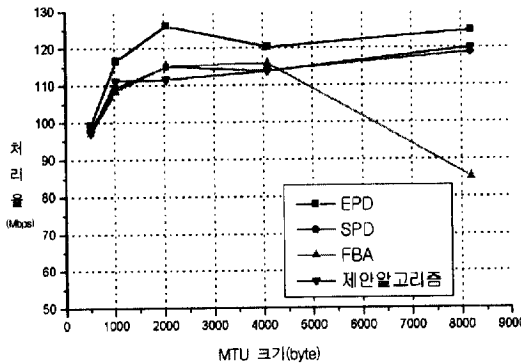
(그림 9) TCP 컨넥션과 스위치 지연에 따른 처리율

(그림 9)와 (그림 10)은 TCP 컨넥션과 스위치 사이의 지연변화 따른 평가로써, VC의 수가 5~70이고 MTU 크기를 1024 bytes, 스위치와 스위치 사이의 지연을 300 time slots로 설정하여, 시뮬레이션하였다. 이러한 시뮬레이션의 목적은 병목 구간 스위치 버퍼에 들어오는 데이터의 지연과 폭주 정도가 처리율과 공정성에 미치는 영향 정도를 조사하기 위함이다. (그림 9)

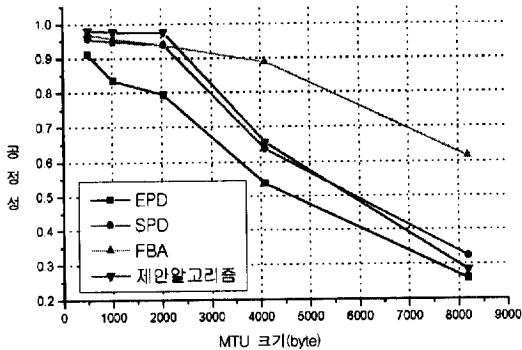
의 처리율에서는 EPD가 다른 알고리즘보다 좋은 성능을 보이고 있다. 그러나 공정성은 제안 알고리즘이 평균 0.979이고 EPD는 0.85, SPD는 0.953, FBA는 0.956을 각각 나타내었다.



(그림 10) TCP 컨넥션과 스위치 지연에 따른 공정성



(그림 11) MTU 크기 변화에 따른 처리율

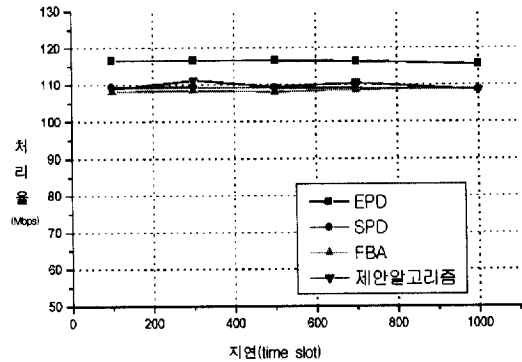


(그림 12) MTU 크기 변화에 따른 공정성

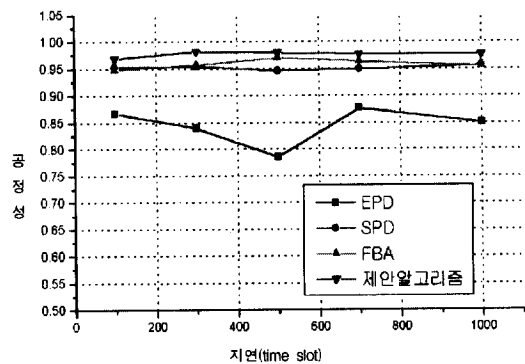
(그림 11)과 (그림 12)는 MTU 변화 따른 평가로써, VC의 수가 5~70이고, TCP 컨넥션과 스위치 지연은

10 time slots, 스위치와 스위치 사이의 지연을 300 time slots로 설정하여 시뮬레이션 하였다.

여기서는 MTU의 크기가 클수록, 결과의 그래프는 많은 변화를 나타내고 있는데, 그 이유로는 스위치 버퍼의 크기가 4096 cell인데 반해 VC의 증가로 인하여 각 VC의 전송되는 데이터 크기의 합이 버퍼의 부하를 높였기 때문이다. EPD, SPD, 제안 알고리즘은 단일 버퍼를 사용하는데 비해, FBA 같은 경우는 VC의 수에 따라 동적으로 가상 버퍼가 생성되어 공정성에서 다른 알고리즘보다 심각한 영향을 덜 받으나, 처리율의 감소는 각 가상 큐에 있는 데이터가 라운드 로빈 방식에 의해 dead 셀을 가지지 않은 패킷을 처리하는데는 다른 알고리즘보다 기다려야 하는 지연이 발생하기 때문이다. (그림 12)에서 제안한 알고리즘의 결과가 MTU 2048 bytes 이상일 경우에는 급격히 공정성이 낮아짐을 알 수 있는데, 문제 해결은 버퍼 크기 증가에 의해 간단히 해결될 수 있다.



(그림 13) 스위치와 스위치 사이의 지연에 따른 처리율



(그림 14) 스위치와 스위치 사이의 지연에 따른 공정성

(그림 13)과 (그림 14)는 스위치와 스위치 사이의 지

연변화 따른 평가로써, VC의 수가 5~70이고 MTU 크기를 1024 bytes, TCP 컨넥션과 스위치 사이의 지연은 10 time slots로 설정하여 시뮬레이션 하였다. (그림 13)의 결과에서 EPD가 다른 알고리즘보다 처리율이 높으며, (그림 14)의 공정성 평가에서는 제안 알고리즘이 평균 0.976 값을 가지고, EPD는 0.844, SPD는 0.951, FBA는 0.958의 값들을 나타내고 있다.

6. 결 론

본 논문에서는 ATM상에서 TCP 성능 향상을 위해, 두 개의 고정된 임계치와 ATM 셀의 VPI 필드 정보를 이용하여 버퍼에서 성공적으로 전송된 횟수에 따라 많이 전송된 셀(Cell)이 있으면 버퍼에서 폐기되고, 적게 전송된 셀일 경우 불필요하게 제거를 당하지 않게 하여 각 노드가 균등한 평균값을 유지하도록 한 방법을 제안하여, peer to peer망 모델에서 여러 가지 알고리즘(EPD, SPD, FBA)과 제안한 알고리즘에 대하여 비교·분석하였다. 또한 제안한 알고리즘의 동작과 새로 추가된 파라미터에 대하여 설명하였다.

결론적으로, VC의 수, 지연, MTU의 변화에 따른 실험 결과로, 제안한 알고리즘은 SPD나 FBA 비교해서 보다 나은 공정성을 보였으며, 처리율면에서도 다른 알고리즘과 동등한 결과를 시뮬레이션을 통해 알 수 있었다.

참 고 문 헌

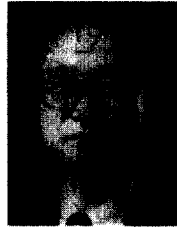
- [1] ATM Forum, ATM Traffic Management Specification, Version 4.0, AF-TM 0056, April 1996.
- [2] D. Kouvatso, "ATM Networks : Performance and modelling and analysis volume 3," 1st ED., Chapman & Hall, 1997.
- [3] Miguel A. Labrador & Sujata Banerjee, "Packet Dropping Policies for ATM and IP Networks," IEEE Communications Surveys, Vol.2, No.3, Third Quarter 1999.
- [4] V. Rosolen, O. Bonaventure, and G. Leduc, "Impact of Cell Discard Strategies on TCP/IP in ATM UBR Networks," Proc. of the 6th. Workshop on Performance Modelling and Evaluation of ATM Network(IFIP ATM'98), UK, July 1998.
- [5] R. Jain et al, Performance of TCP over UBR and Buffer Requirements, ATM-Forum/96-0518, April 1996.
- [6] S. Kalyanaraman, Raj Jain, Sonia Fahmy, Rohit Goyal, "Performance and Buffering Requirements of Internet Protocols over ATM ABR and UBR Services," IEEE Communications Magazine, pp. 152-157, June 1998.
- [7] A. Romanow, S. Floyd, "Dynamics of TCP Traffic over ATM Networks," IEEE Journal on Selected Areas in Communications, Vol.13, No.4, pp.633-641, May 1995.
- [8] H. Li, K.-y. Siu, H.-y. Tzeng, C. Ikeda, H. Suzuki, "A Simulation Study of TCP Performance in ATM Networks with ABR and UBR Services", Proceedings of IEEE INFOCOM'96, Vol.3, pp.1269-1276, March 1996.
- [9] G. Hasegawa, H. Ohsaki, M. Murata, H. Miyahara, "Performance Evaluation and Parameter Tuning of TCP over ABR Service in ATM Networks", IEICE Transactions on Communications, Vol.E79-B, No 5, pp.668-682, May 1996.
- [10] R. Goyal, R. Jain, S. Kalyanaraman, S. Fahmy, and S. Kim, "UBR+ : Improving Performance of TCP over ATM-UBR Service," Proc. of IEEE ICC, pp. 1042-1048, 1997.
- [11] Heinanen J, Kilkki K, "A fair buffer allocation scheme," Computer Communications, V.21, pp. 220-226 January 1998.
- [12] YATS simulator for ATM networks, Dresden University of Technology, 1997.
- [13] L. Jaussi, M. Lorang, J. Nelissen, "A Detailed Experimental Performance Evaluation on TCP over UBR," IEEE ICATM '98, pp.214-223, June 1998.
- [14] Bobby Vandalore, Sonia Fahmy, Raj Jain, Rohit Goyal, Mukul Goyal "General Weighted Fairness and its Support in Explicit Rate Switch Algorithms," Computer Communications, Vol.23, No.2, pp.149-161, January 2000.
- [15] V. Jacobson, "Congestion Avoidance and Control," Proceedings of the SIGCOMM '88 Symposium, pp. 314-332, August 1988.



육 동 철

e-mail : net607@mail1.pknu.ac.kr
1998년 동서대학교 컴퓨터공학과
졸업(공학사)
2000년 부경대학교 전자계산학과
졸업(이학석사)
2001년~현재 부경대학교 전자계산
학과 박사과정

관심분야 : 컴퓨터 네트워크 및 시뮬레이션, 이동통신



박 승 섭

e-mail : parkss@dolphin.pknu.ac.kr
1982년 경북대학교 공과대학 전자
계산전공(공학사)
1984년 일본대학 이공학연구과
(공학석사)
1993년 일본 동북(Tohoku)대학
(공학박사)

1984년~1986년 한국통신 연구원
1989년~1990년 일본동북대학 객원 교수
1998년 Philippine Ateneo de davao university, visit-
ing prof.
1986년~현재 국립 부경대학교 전자컴퓨터정보통신공학
부 교수
1998년~1999년 부경대학교 컴퓨터멀티미디어 공학부
학부장
관심분야 : 인터넷 엔지니어링, 초고속통신망, 멀티미디
어통신