

ATM 서브망에서 MCS 멀티캐스트 구현을 위한 전송 제어 시스템의 성능 평가

Analysis of Traffic Control System for Supporting MCS Multicasting on ATM Subnetworks

박 상 준*, 이 효 준*, 김 관 중*, 김 영 한**, 김 병 기*

(Sang-Joon Park*, Hyo-Jun Lee*, Kwan-Joong Kim*, Young-Han Kim**, Byung-Gi Kim*)

요 약

ATM 망으로 연결된 하나의 서브 네트워크에서 MCS(Multicast Server)에 의한 멀티캐스팅은 전송 폭주에 대해 효율적인 제어방식을 요구한다. 본 논문에서는 MCS를 이용하여 TCP 패킷에 대한 멀티캐스팅을 구현하고, 버퍼의 효율적 전송 제어를 위해 공통 버퍼를 가지는 EPD +SPD방식(Early Packet Discard-same Source Packet Discard)을 제안한다. MCS에서 전송 버퍼에 대한 임계치를 넘어서면 버퍼에 오버플로우가 발생하기 이전에 셀을 폐기시키는 것으로 손상된 패킷의 전송을 막기 위하여 부분적 셀 폐기 대신 손상된 패킷 전체를 폐기하는 방식(EPD-Early Packet Discard)을 제공한다. 또한 TCP 패킷의 재전송 중복에 대한 부하를 줄이기 위해 동일한 송신측의 패킷을 폐기하는 방식(SPD-same Source Packet Drop)을 제안한다. 시뮬레이션 결과에서는 전송 데이터 폐기에 대한 기존의 방식(Tail Drop), EPD방식을 고려한 DFF(Drop From Front)와 본 논문에서 제안한 EPD +SPD방식과의 성능 분석을 한다.

ABSTRACT

The multicasting of MCS(Multicast Server) requires a effective traffic control scheme to prevent buffer overflow on ATM subnetworks. This paper considers MCS multicasting to TCP packets, and propose EPD +SPD scheme(Early Packet Discard-same Source Packet Discard) using common buffer. When the threshold of output buffer is reached, MCS drops an entire packet prior to buffer overflow, so that corrupted packets will not be transmitted by the server. And SPD scheme will be used to decrease the overload of TCP packet re-transmission by discarding same source packet. Our simluations show that the EPD +SPD results in higher TCP throughput than that of tail drop and EPD +DFF.

I. 서 론

최근 ATM 망에서 분배적 전송 서비스나 영상회의와 같은 그룹웨어 응용들을 지원하기 위해 다자간 통신의 연구가 활발히 진행 되고 있다. 멀티캐스트는 하나의 송신측이 그 이상의 수신측에게 데이터를 동시에 전송하는 것을 의미한다[1]. ATM Forum에서는 MPOA 네트워크에 존재하는 호스트들 사이에서 AAL_SDU의 멀티캐스트와 브로드캐스트를 구현하기 위해 ATM 연결에 참여하는 호스트 집합을 클러스터(cluster)로 정의하고 있다[2]. 인트라-클러스터(intra-cluster)에서 멀티캐스트 구현을 위한 전송모는 송신측의 AAL_SDU를 다중 목적지에 전송

하는데 쓰인다. 송신측은 AAL_SDU를 멀티캐스트하기 위해 전송 작용을 하게 되는데 VC를 통하여 UNI에 전송된다. UNI 3.1은 송신측의 전송을 멀티캐스트 전송(point-to-multipoint VC)으로 하기 위한 인터페이스를 제공한다. ATM Forum에서 제공하는 멀티캐스트 방식은 크게 두 가지로 나누어진다. 하나가 멀티캐스트 VC 메쉬(Mesh) 방식이며 다른 하나가 멀티캐스트 서버(MCS)를 두는 방식이다. 멀티캐스트 메쉬 방식은 송신측이 다중 수신측에 대해 각기 VC 연결을 하고 있어서 멀티캐스팅을 할 때 전송하고자 하는 목적지의 VC에 데이터를 보내주는 방식이다[9].

MCS 방식은 한 클러스터 안에 멀티캐스트를 위한 서버를 두어 송신측이 멀티캐스트를 원할 때에 멀티캐스트용 서버에게 데이터를 전송하여 수신측에 전달하는 방식이다. 멀티캐스트 메쉬 방식은 각 호스트에 대한 VC 연결로 인한 비용이 많이 드며, MCS 방식은 서버에 대한

* 숭실대학교 전자계산과
** 숭실대학교 정보통신공학과
접수일자: 1997년 4월 8일

전송 부하가 크다는 단점이 있다.

본 논문에서는 TCP 멀티캐스트 구현을 위해 MCS를 이용하며, 효율적인 대역폭 사용을 위해 셀을 패킷 단위로 폐기하는 EPD 방식을 사용하였다. 그리고 패킷의 폐기시에 재전송 부하를 고려하여 동일한 송신측의 패킷을 폐기하는 방식(SPD)을 제안하였다. 기존의 Tail Drop, EPD + DFF 방식(Early Packet Discard-Drop From Front)과 EPD + SPD방식의 성능을 시뮬레이션으로 비교 분석하였다.

II. EPD(Early Packet Discard)와 DFF(Drop From Front)

EPD 방식은 전송 버퍼의 폭주시에 낭비되는 대역폭을 효율적으로 관리하기 위해 제시된 방식이다[3, 5]. TCP상에서 전송 폭주에 대한 기존의 ATM 셀 단위의 폐기는 손상된 패킷을 네트워크에 전송한다는 측면에서 대역폭 낭비를 가져왔다. 즉, 스위치 버퍼가 차게되면 전송된 셀을 폐기하게 되는데, 여기서 폐기된 셀이 속한 TCP 패킷이 수신측에 그대로 전달되게 된다. 네트워크 상에서 이렇게 손상된 패킷에 대한 전송은 대역폭의 낭비를 가져오며 재전송에 대해 그만큼의 지연을 초래한다.

EPD는 각 스위치 버퍼에 임계치를 두는 방식으로 구현된다. 스위치 버퍼에 도착된 셀이 임계치를 넘어서면 셀을 폐기하게 되는데, 단순히 도착되는 셀만 폐기하는 대신 패킷 전체를 폐기하여 뒤에 도착되는 셀들의 진입 공간을 제공하며, 손상된 패킷을 미리 제거하는 것이다. EPD 임계치의 값은 [7]에서 제시하였으며 아래와 같다.

$$EPD\ Threshold = 0.5 \times Output\ Buffer\ Size$$

EPD방식은 DFF방식과 같이 쓰일 수 있다. DFF방식은 전송 폭주 시에 TCP 패킷의 폐기를 앞에서 하여 패킷 폐기에 대한 송신측의 빠른 대응을 유도할 수 있다. TCP 송신측은 폭주에 대한 패킷의 폐기와 전송 사이즈를 조절하여 재전송에 대해 신속히 대처할 수 있다[6, 10]. 그림 1은 유니 캐스트 전송상에서 스위치의 출력 버퍼에 발생한 폭주를 EPD + DFF 방식이 처리하는 과정을 보이고 있다.

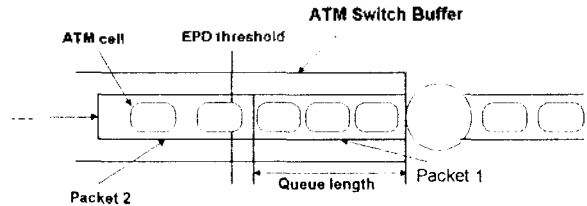
III. 제안된 EPD + SPD(same Source Packet Discard) 방식

본 논문에서는 MCS를 기반으로하는 멀티캐스트 지원 방식에 대하여 연구한다. 현재 TCP/IP에서 연결지향의 TCP는 전송제어 메카니즘을 포함하며, 멀티캐스트 지원을 위한 비연결형의 UDP는 폐기된 패킷에 대해 재전송을 하지 않는다[8]. 본 장에서는 이러한 TCP의 연결지향적인 전송제어 메카니즘에 UDP의 멀티캐스트 특성을 결합하여 멀티캐스팅 기능을 제공하는 MCS에 관해 기술한

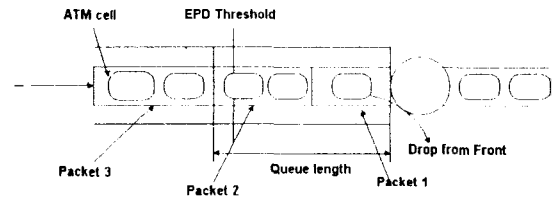
다. 그리고 ATM 망의 일대일 연결상(Point-to-Point)에서 효율적인 TCP 패킷 전송을 위해 제공된 EPD 방식과 여러 송신측의 재전송 중복을 피하기 위해 제안된 SPD 방식을 설명한다.

3.1 공통 버퍼

송신측은 TCP를 변경하지 않고 MCS에 VC 연결을 통하여 멀티캐스트 패킷을 ATM 셀(AAL_SDU)로 서버에게 전달한다. 송신측으로부터 멀티캐스트 패킷을 받은 MCS는 여러 수신측에 패킷을 전달하기 위해 각각의 스위치의 출력 버퍼에 복사된 패킷을 전송한다. 그림 2는 이러한 송수신측과 MCS와의 관계를 보이고 있다. MCS에는 공통 버퍼가 있으며 멀티캐스트 패킷을 받으면 우선 공통 버퍼에 저장한다. 지속적으로 멀티캐스트 패킷이 전송되면 공통버퍼에 저장되고 공통버퍼가 차게되면 처음 저장된 패킷은 폐기된다.



(a) 패킷의 버퍼링



(b) 패킷 폐기(Drop From Front)

그림 1. EPD + DFF

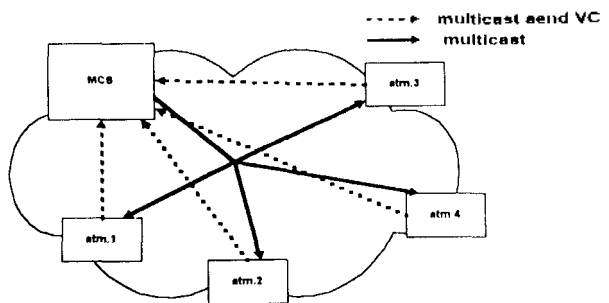


그림 2. Multicast Server

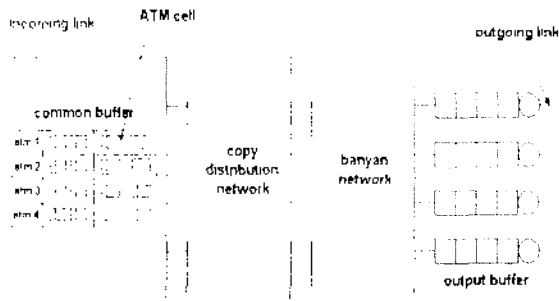


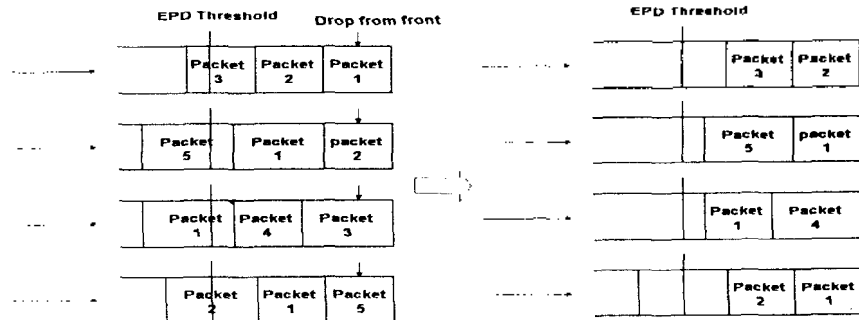
그림 3. MCS의 구조와 동작

서버는 멀티캐스트 패킷을 공통버퍼에 저장한 후, 수신측의 주소대로 패킷을 복사한 후 해당 스위치 출력 포트의 버퍼에 전달한다. 그림 3에 보는 바와 같이 ATM셀로 전송된 TCP 패킷은 MCS에서 다시 TCP 패킷으로 재조립되고, 해당 출력 큐에 전달된다.

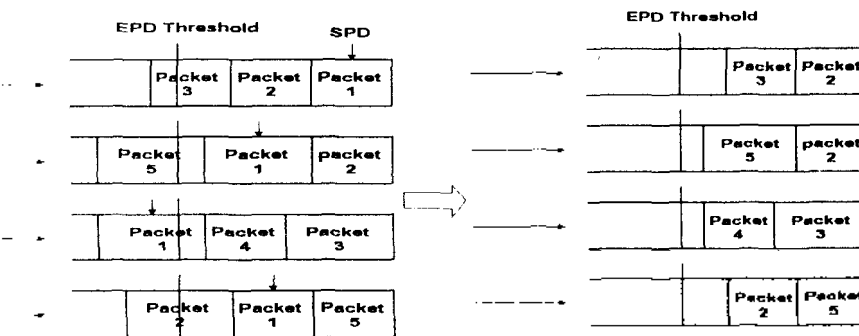
3.2 EPD + SPD(same Source Packet Discard)

EPD는 유니캐스트상에서 효율적인 TCP 패킷 전송을 위해 제안된 메카니즘이다. 멀티캐스트 서비스를 위해 MCS를 이용하는 본 연구에서도 TCP 패킷의 전송방식으로 EPD 메카니즘을 이용하며 중복적 재전송을 막기 위해 SPD방식을 제시한다. 만일 MCS의 여러 스위치 출력 버퍼에 폭주가 발생하였다면 EPD 방식에 의해 같은 TCP 패킷에 속한 ATM 셀은 폐기된다. 하지만 패킷의 폐기시

에 DFF와 같이 큐의 앞부분의 패킷이 무조건 폐기되는 것이 아니라 여러 출력 큐에서 같은 송신측의 패킷이 폐기된다. 본 논문에서는 이렇게 송신측이 같은 패킷을 폐기하는 방식을 SPD(same Source Packet Discard)라 정의한다. 이렇게 되면 여러 개의 패킷에 대해 재전송이 중복적으로 발생하는 것을 막을 수 있으며, 재전송에 대한 복잡성 및 지연을 방지할 수 있다. 또한 여러 송신측의 재전송으로 인해 발생할 수 있는 전송 폭주의 재발을 막을 수 있다. 그림 4에서 DFF와 SPD의 동작을 비교하여 보이고 있다. 그림 4(a)의 DFF 방식에서는 여러개의 패킷이 폐기되므로 네 개의 패킷(1, 2, 3, 5)들이 재전송되어야 하며, 그림 4(b)의 SPD에서는 패킷(1)만 재전송되면 된다. 폐기될 패킷은 서버에 의해 관리되는데 송신측의 패킷들은 라운드-로빈에 의해 폐기된다. 그림 5는 SPD 메카니즘을 사용한 멀티캐스트의 한 예이다. 그림 A의 각 송신측에서 그룹 B의 수신측에 멀티캐스트 패킷을 전송한다고 할 때, 송신측 (atm1, atm2, atm3, atm4, atm5)이 MCS에 데이터를 전달한다. 만약 MCS의 스위치 버퍼에 폭주가 발생하였다면, 먼저 각 큐에서 atm1의 셀들(같은 패킷에 포함됨)을 폐기한다. 폐기 후에도 폭주가 해결되지 않으면 라운드 로빈 방식에 의해 atm2의 셀들을 폐기한다. 이런 방식으로 버퍼의 임계치 값보다 작아질 때까지 폐기를 계속한다. 만약 스위치의 버퍼에서 폭주가 누그러지면 MCS는 폐기된 해당 패킷을 공통버퍼에서 다시 출력 버퍼에 전송하며, 공통 버퍼에 없을 시에는 송신측에



(a) EPD + DFF 방식



(b) EPD + SPD 방식

그림 4. TCP 패킷의 폐기 방식

다 재전송요구를 한다.

스위치 버퍼의 폐기로 인한 요구이외에 외부적 문제로 재전송 요구가 발생할 수 있다. 본 논문의 공통버퍼 방식은 외부에서 재전송요구가 올 경우 먼저 서버의 공통버퍼에서 해당 재전송 패킷을 찾는다. 그림 6은 EPD + SPD 방식의 제어 알고리즘이다.

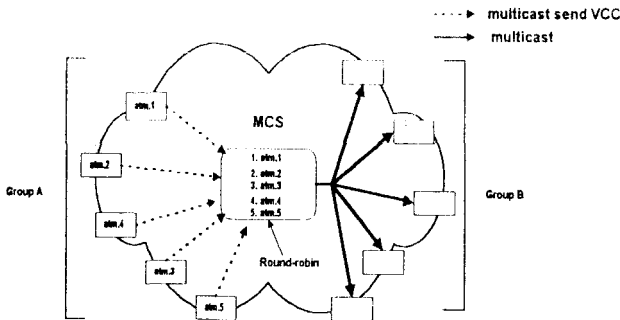


그림 5. MCS의 데이터 전송

```

/* output buffer control for TCP packets*/
if connection type = TCP connection type
/* queue threshold = 0.5 * max_buf_size */
if output queue length < queue threshold
    add TCP packet into output queue
else
    output queue congested = TRUE
    /* select source for discard through round robin */
    select a source packet to discard
    discard a TCP packet of the source from front buffer
    the discarded source count += 1
    discarded packet count += 1
    
```

그림 6. EPD + SPD 방식의 제어 알고리즘

만약 해당 패킷이 공통버퍼에 있을 경우 MCS에서 자동적으로 재전송되며 공통버퍼에 없을 경우 송신측에다 재전송요구를 보낸다. 본 논문에서 외부 재전송 요구에 대한 재전송 방식은 서버에서 송신측에 재전송을 요구하는 것과 기존의 TCP(BSD 4.3 Tahoe version)에서 사용된 재전송 방식을 채택하였다.

IV. 시뮬레이션 환경 및 분석

본 장에서는 시뮬레이션을 통해 기존의 방식(Tail drop), EPD + DFF와 제안된 방식(EPD + SPD)의 성능을 비교한다.

4.1 시뮬레이션 환경

그림 6에서는 본 논문에서 사용된 시뮬레이션의 망 환

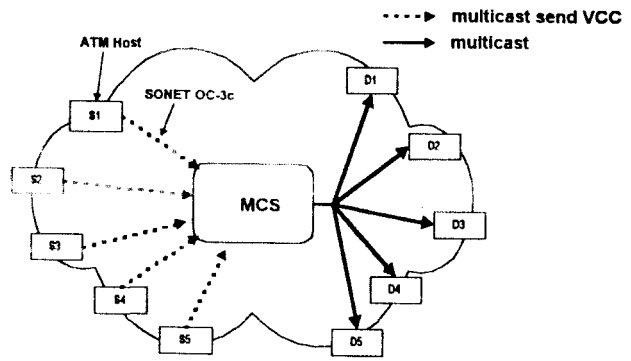


그림 7. 시뮬레이션 환경

경이다. 각 ATM 호스트와 MCS와의 연결은 SONET OC-3c 링크로 되어있다. TCP 트래픽은 ATM 망을 통하여 전달되며 BSD 4.3 Tahoe 버전으로 한다. TCP 패킷의 크기는 512 byte와 8KB로 하며 TCP 윈도우 사이즈는 8KB(Max Receiver Window Size)로 한다. 각 송신측은 동일한 크기의 데이터를 전송하며, 출력 큐잉의 폭주가 발생할 때 까지 TCP 윈도우 사이즈는 증가한다. TCP 윈도우 크기의 조절은 윈도우 기반 전송 제어 방식(window-based flow control mechanism)을 따른다. 공통 버퍼의 크기는 각 송신측의 윈도우 사이즈의 5배로 하였다. 스위치의 버퍼 크기는 500, 1000, 2000, 3000과 4000 셀로 한다. 또한 버퍼의 임계치는 큐 크기의 반으로한다. MCS의 스위치는 16x16 출력 버퍼를 가지며, 각 출력 큐의 셀 진입은 FIFO로한다. 각 송신측은 수신측에 멀티캐스팅을 하며, 전송 서비스는 동일한 수신측에 데이터를 전송하는 상호 대화형 서비스로한다. 각 component를 연결하는 링크에 대한 전송지연은 5μ로한다. TCP 패킷의 재전송은 timeout에 의해 발생할 수 있으며 서버에서 재전송 요구를 할 수 있다. 처음 하나의 TCP 패킷을 윈도우 크기로 하며, ACK 신호가 올때마다 증가시켜 데이터를 전송한다. 만약 스위치 버퍼의 임계치를 넘었다면 폭주가 발생한 것으로 인식 폭주 회피모드로 데이터를 전송한다. 재전송모드에서는 전송 윈도우 사이즈를 다시 한 개의 TCP패킷으로 한다. 외부에 대한 재전송 요구는 먼저 공통버퍼에 있는지 검색하여 해당 패킷이 있다면 출력 버퍼로 전송하고 없으며 송신측에 재전송 요구를 한다.

4.2 시뮬레이션 결과 분석

그림에서 시뮬레이션 결과는 송신측의 전송에 대한 'effective throughput'과 단위시간당 재전송 횟수를 나타내었다. 그림 8은 TCP 패킷의 크기를 512 byte와 8KB로 했을 때의 각 전송원의 throughput을 평균하여 보이고 있다. 그림 8의 (a)와 (b)에서 알 수 있듯이 MCS 출력 버퍼의 크기를 변화시켰을 때 EPD + SPD방식이 Tail Drop과 EPD + DFF방식에 비해 효율적인 성능을 보이고 있다. 그림 9의 (a)와 (b)는 버퍼 크기를 1000 셀로 하였을

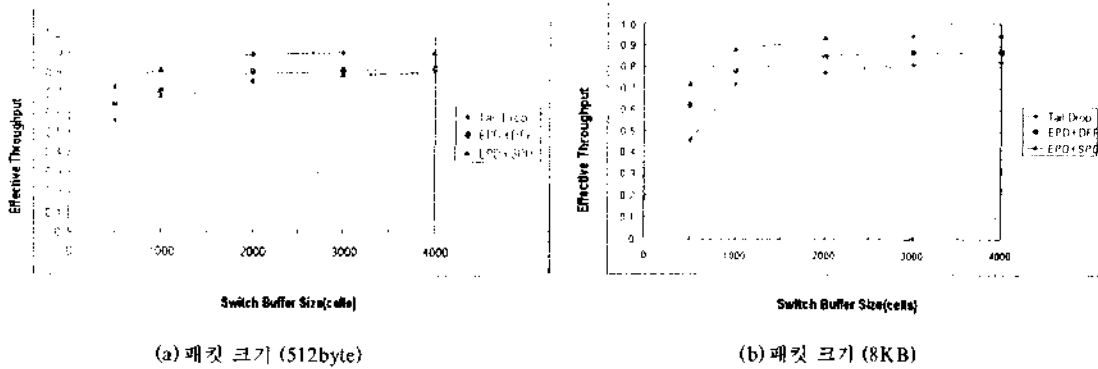


그림 8. 버퍼 크기에 따른 effective throughput

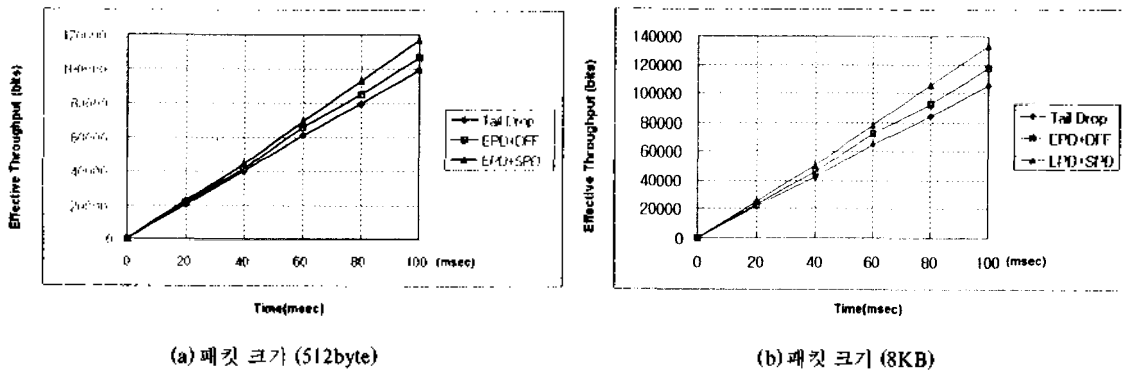


그림 9. 시간의 흐름에 따른 throughput

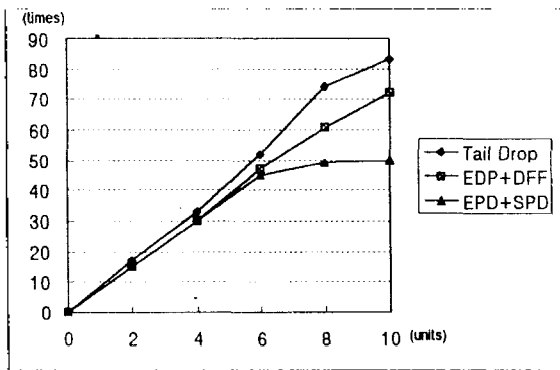


그림 10. 단위 시간에 따른 재전송 횟수

때 시간의 흐름에 딸 데이터의 효율적 전송에 대해 비교하고 있다. 그림 10에서는 단위 시간당 NAK 신호로 인한 TCP 패킷의 재전송 횟수를 보이고 있다. EPD + DFF 방식의 경우 Tail Drop 보다 재전송에 대한 부하가 적은 반면 전송 폭주에 대한 앞단의 임의 패기로 인해 손실에 대한 지속적인 증가를 보이고 있으며 EPD + SPD 같은 경우 재전송에 대한 안정을 보이고 있다. EPD + SPD 방식의 경우 EPD + DFF 방식에 비해 소프트웨어적으로 버퍼를 처리할 경우 그 과정이 복잡하게 되므로, 하드웨어적인 제어 방식이 요구된다.

V. 결 론

본 논문에서는 ATM 서브망에서 MCS를 이용한 TCP 패킷의 멀티캐스팅 방식을 제안하였으며, 이에 대한 성능을 시뮬레이션을 통하여 분석하였다. 버퍼에 대한 제어는 망의 효율성을 위해 EPD 방식을 사용하였고 재전송을 고려하여 SPD 방식을 제안하였다. 시뮬레이션 결과에서는 본 논문에서 제시한 EPD + SPD 방식이 기존의 Tail Drop과 EPD + DFF 방식에 비해 우수한 성능을 보였으며 재전송에 대한 안정을 유지하였다. 향후 연구는 EPD + SPD에서 DFF 방식을 고려한 EPD + DFF + 방식의 연구가 진행될 것이며 멀티캐스팅의 효율적 재전송 메커니즘이 고려될 것이다.

참 고 문 헌

1. V.P. Kompella, J.C. Pasquale and G.C. Polyzos, "Multicast Routing for Multimedia Communication," IEEE/ACM Transactions on Networking, Jan, 1993.
2. Grenville Armitage, "support for Multicast over UNI 3.1 based ATM Networks," INTERNET-DRAFT <draft-ietf-ipatm-04.txt>, IP over ATM Working Group, February, 1995.
3. Allyn Romanow and Sally Floyd, "Dynamics of TCP Traffic over ATM Networks," IEEE JSAC, pp 633-641, VOL. 13, NO. 4, MAY 1995.

4. T. V. Lakshman, Arnold Neidhardt and Teunis J. Ott, "The Drop from Front Strategy in TCP and in TCP over ATM," IEEE INFOCOM, pp 1242-1250. March 1996.
5. Hongqing Li, Kai-yeung Siu and Hong-Yi Tzeng, "A Simulation Study of TCP Performance in ATM Networks with ABR and UBR Services," IEEE INFOCOM, pp 1269-1276. March 1996.
6. Jacobson, V., "Congestion Avoidance and Control," Proceeding of the ACM SIGCOMM'88, August 1988.
7. D. Sisalems, "Rate Based Congestion Control and its Effects on TCP over ATM." May, 1994 <http://ptolemy.eecs.berkeley.edu/papers/tcpSim/>
8. D. Cavendish, S. Mascolo, and M. Gerla, "Rate Based Congestion Control for Multicast ABR Traffic," IEEE GLOBECOM'96, pp 1114-1118, November 1996.
9. Rajesh R. Talpade, Grenville J. Armitage, and Mostafa H. Ammar, "Experience with Architecture for Supporting IP Multicast over ATM," IEEE ATM'96, Volume 1, August 1996.
10. Wenge Ren, Kai-yeung Siu, and Hiroshi Suzuki, "On the Excess Buffer Capacity for Early Packet Discarding Schemes in ATM," IEEE ATM'96, Volume 2, August 1996.
11. Raif O. Onvural, "Asynchronous transfer mode networks: performance issues," Boston: Artech House, second edition, 1995.

▲박 상 준(Sang-Joon Park)



1996년 2월: 동국대학교 전자계산학과(학사)
 1996년~현재: 숭실대학교 대학원 전자계산학과(석사과정)
 ※주관심분야: ATM 트래픽 제어, 이동무선통신, 위성통신 시스템 등

▲이 효 준(Hyo-Jun Lee)



1996년 2월: 숭실대학교 전자계산학과(학사)
 1996년~현재: 숭실대학교 대학원 전자계산학과(석사과정)
 ※주관심분야: ATM 트래픽 제어, 멀티캐스팅 프로토콜, 이동무선통신 등

▲김 관 중(Kwan-Joong Kim)



1983년 2월: 숭실대학교 전자계산학과(학사)
 1988년 8월: 숭실대학교 대학원 전자계산학과(석사)
 1993년 2월: 숭실대학교 대학원 전자계산학과 박사과정 수료
 1990년 9월~1996년 8월: 숭실대학교 정보과학대학 컴퓨터학

부 시간강사

1997년 3월~현재: 한서대학교 전산학과 전임강사
 ※주관심분야: 컴퓨터구조, 컴퓨터통신, 병렬처리 등

▲김 영 한(Young-Han Kim)



1984년 2월: 서울대학교 전자공학과(학사)
 1986년 2월: 한국과학기술원 전기및전자공학과(석사)
 1990년 8월: 한국과학기술원 전기및전자공학과(박사)
 1987년~1994년 2월: 디지털 정보통신 연구소, 데이터통신 연구부장

1994년~현재: 숭실대학교 정보통신공학과 조교수
 ※주관심분야: ATM 트래픽 제어, B-ISDN, 인터넷 멀티미디어통신, 이동 멀티미디어통신 등

▲김 병 기(Byung-Gi Kim)



1977년 2월: 서울대학교 전자공학과(학사)
 1979년 2월: 한국과학기술원 전산학과(석사)
 1997년: 한국과학기술원 전산학과(박사)
 1979년~1982년: 경북대학교 전자공

학과 전임강사
 1982년~현재: 숭실대학교 정보과학대학 컴퓨터학부 교수
 ※주관심분야: 컴퓨터구조, 컴퓨터통신, 병렬처리 등