IE Interfaces
Vol. 15, No. 2, pp. 152-158, June 2002.

VC 머징이 가능한 ATM LSR의 구조 및 트래픽 엔지니어링 연구

정호연[1] · 서재영[2] · 백장현[2†]

[1]전주대학교 산업공학과 / [2]전북대학교 산업시스템공학과

Study on Architecture of ATM LSR Supporting VC Merging and Traffic Engineering over It

Ho-Yeon Chung[1] · Jae-Young Seo[2] · Jang-Hyun Baek[2]

[1] Department of Industrial Engineering, Jeonju University, Jeonju, 560-759

[2] Department of Industrial and System Engineering, Chonbuk National University, Jeonju, 561-756

The explosive growth of the Internet traffic in the last few years has imposed tremendous stress on today's routers, particularly in the core network. Recently, ATM LSRs(Label Switching Router) are potentially capable of providing the highest forwarding capacity in the backbone Internet network. VC merging is a mechanism in an ATM LSR that allows many IP routes to be mapped to the same VC label, and provides a scalable mapping method that can support thousands of destinations. VC merging requires reassembly buffers so that cells belonging to different packets intended for the same destination do not interleave with each other.

In this study, we propose an architecture of the ATM LSR which supports VC merging. We propose traffic control scheme called APD(Active Packet Discard) algorithm so that predicts and controls the congestion of the Internet traffic effectively. We study the performance of this algorithm using simulation.

Keywords: VC merging, label switching router, traffic control

1. 서 론

최근 정보화 사회로의 급속한 변화와 기술 발달 등에 힘입어 인터넷 사용자가 폭발적으로 증가하고 있다. 또한 비실시간 데이터 전송 위주이던 인터넷 서비스에서 고속의 대역폭을 필요로 하는 멀티미디어 서비스로의 변화가 급속히 확산되고 있다. 이러한 인터넷 트래픽 환경의 급속한 변화에 대처하기 위하여 IETF (Internet Engineering Task Force)는 비연결형 인터넷망에 연결형 메커니즘을 도입한 MPLS (Mutiprotocol Label Switching) 개념을 제안하였다. 스위칭과 라우팅의 통합 모델로서 제시된 MPLS 기술은 패킷 포워딩 과정에서 IP 패킷 헤더를 직접 이용하지 않고, 고정된 짧은 길이의 레이블(label)을 이용하여 패킷 포워딩을 용이하게 해준다. MPLS 기술은 <그림 1>과 같이 기존의 패킷 포워딩 기술 대신에 라우팅 정보를 짧은 길이의 레이블로 맵핑하여 계층 2에서 IP 패킷 스위칭이 이루어지도록 함으로써 인터넷 백본망에서 야기될 수 있는 대역폭 부족 문제를 해결할 수 있을 것으로 기대되고 있다(Callon, Swallow, Feldman, Viswanathan, Doolan and Fredette, 1999; Callon, *et. al.*, 1999).

이 논문은 2000년도 한국학술진흥재단의 지원(KRF-00-E00130)에 의하여 연구되었음.
† 연락저자 : 백장현 교수, 561-756 전북 전주시 덕진동 1가 664-14 전북대학교 산업시스템공학과, Fax : 063-270-2333, e-mail : jbaek@moak.chonbuk.ac.kr
2002년 2월 접수, 1회 수정 후 2002년 3월 게재 확정.

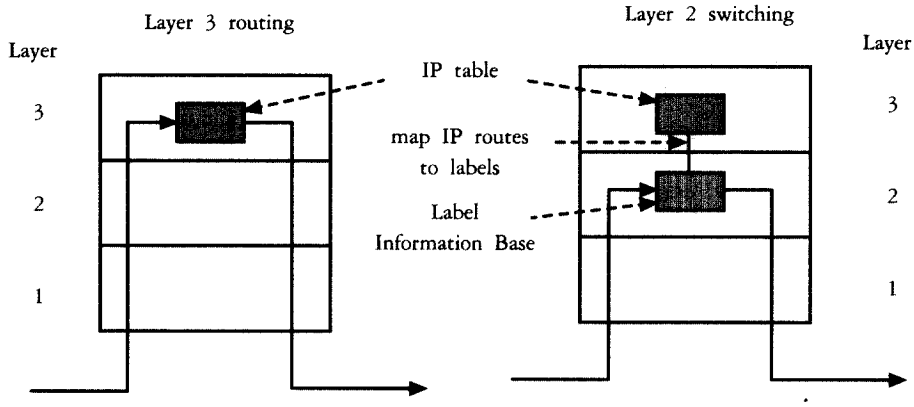


그림 1. 계층 3 라우팅과 계층 2 스위칭.

이러한 MPLS 환경에서는 ATM의 고속 스위칭 기능이 인터넷 백본망에 대해 효율적인 전송을 제공할 것으로 기대되고 있다. 그러나 기존의 ATM망은 점대점 연결과 점대다중점 연결을 지원하고 있으나 다중점대점 연결 기능을 제공하지 못하고 있다. N개의 라우터간 상호 연결을 위해서는 점대점 연결이 N^2개가 필요한 반면 다중점대점 연결은 N개가 필요하다. 따라서 다중점대점 연결은 연결 수를 현격히 줄일 수 있다. 이 방법은 레이블 수를 줄이고 확장성을 용이하게 하는데, MPLS에서는 이러한 기술을 머징(merging)이라고 한다. 다중점대점 연결을 구현하기 위한 방법으로는 VP 머징(VP merging)과 VC 머징(VC merging) 방법이 있다(Chow, Leon-Garcia, 1999; Widjaja and Elwalid, 1999).

VP 머징의 경우, 여러 개의 입력 VP들을 하나의 VP로 병합시키고 다른 송신원으로부터의 트래픽들에게는 VP 내의 다른 VCI 값들을 할당하여 각각의 송신원을 구별한다. 그러나 VP 머징은, ATM에서 VPI 수가 4096개로 제한되어 있으므로 확장성에 문제가 있으며 전체적으로 VCI를 할당하고 운용하기 위한 별도의 프로토콜이 필요하다는 단점이 있다.

VC 머징은 하나의 동일한 VPI/VCI 값으로 여러 개의 입력 VPI/VCI를 병합하는 방법이다. VC 머징은 다중점대점 연결에서 각각의 송신원이 링크당 하나의 VCI를 사용하므로 확장성 면에 있어서 VP 머징보다 우수하므로(Chow, Leon-Garcia, 1999), ATM에서는 VC 머징이 가능한 LSR(Label Switching Router)을 포함하는 구조로 하여 인터넷 트래픽을 수용하려는 추세이다 (Keith McCloghrie *et. al.*, 1999; Baldi Mario, Davide Bergamasco, Silvano Gai, Dante Malagrin, 1999).

ATM 교환기는 입력 VCI를 해당하는 출력 VCI로 변환하여 전달하므로, VC 머징의 경우에는 다른 인터페이스로부터 입력되는 다른 패킷에 속하는 셀들이 섞이는 셀 인터리빙(interleaving)이 발생한다. 각 패킷에 속하는 셀들을 구별해 내기 위하여 패킷 재구성(reassembly)이 병합 지점(merged point)에서 수행되어야 한다. 다중점대점 연결의 모든 패킷에 대한 패킷 재구성을 위하여 병합 지점에서는 추가적인 버퍼가 요구된다. 병합 지점에서는 송신원으로부터 수신된 패킷의 EOP(End Of Packet) 셀이 도착할 때까지 셀들을 버퍼에 저장하여 패킷별로 전달한다. 이와 같이 VC 머징을 위해서는 하나의 패킷이 다 도착할 때까지 버퍼에 셀들을 저장해야 하므로 일반적으로 버퍼 용량이 증가하고 셀 지연이 증가하게 된다(Iliadis and Droz, 1999).

ATM LSR에서 VC 머징을 구현하고자 할 경우 발생하는 이러한 문제점을 완화하기 위해서 본 논문에서는 VC 머징이 가능한 LSR의 체증 제어를 위하여 기존의 방법을 개선한 APD (Active Packet Discard) 알고리듬을 제안한다. APD 알고리듬에서는 이중 지수평활(D-EWMA, Double Exponential Weighted Moving Average) 방법을 통한 low-pass filter를 사용한다. 과거와 현재의 트래픽 상태에 대한 정보를 수집, 저장하고 이러한 정보를 이용하여 체증 현상이 발생할 경우 이후 도착하는 셀에 대한 폐기를 능동적으로 수행하도록 설계하였다.

서론에 이어 2장에서는 VC 머징이 가능한 ATM LSR의 구조에 대하여 기술하고 3장에서는 ATM LSR의 인터넷 트래픽에 대한 체증 제어 방법으로 제안한 APD 알고리듬을 기술한다. 또한 4장에서는 시뮬레이션을 이용하여 트래픽 관련 사항들에 대한 성능평가를 수행하여 결과를 제시한다.

2. VC 머징이 가능한 ATM LSR의 구조

VC 머징이 가능한 ATM LSR에 대한 중요한 요구사항은 서로 다른 패킷에 속한 셀들이 동일한 VC로 합해지더라도 서로 섞이지 않도록 해주어야 한다는 것이다. 이를 위해서는 각각의 패킷에 속한 셀들이 그 패킷의 마지막 셀이 도착할 때까지 재결합 버퍼(RB, Reassembly Buffer)에 저장되어 있다가 마지막 셀이 도착하면 비로소 패킷에 속한 모든 셀들이 다음 노드로 전달되기 위한 출력 버퍼(OB, Output Buffer)로 전달되어야 한다. 이러한 재결합 버퍼는 ATM 스위치의 입력단과 출력단 어느 쪽에나 위치할 수 있다.

재결합 버퍼를 ATM 스위치의 입력단에 두는 구조를 입력 버퍼(Input buffer) 방식이라 하고 재결합 버퍼를 ATM 스위치의 출력단에 두는 구조를 출력 버퍼(Output buffer) 방식이라 한다.

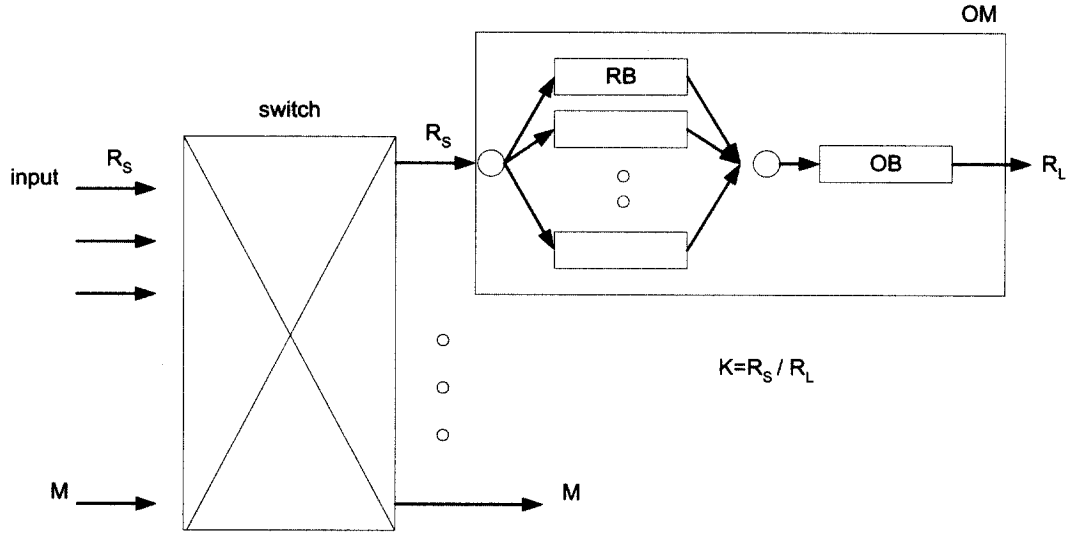


그림 2. VC 머징이 가능한 ATM LSR의 구조.

입력 버퍼 방식의 경우, 즉 재결합 버퍼가 입력단에 위치하는 경우에는 ATM 스위치가 패킷에 속한 모든 셀들을 모아서 보내기 위해서는 프레임 스위칭(frame switching)을 수행해야만 한다. 그런데 프레임 스위칭은 유연하지 못할 뿐만 아니라 여러 가지 QoS가 요구되는 환경에서는 심한 HOL(Head Of Line) 블록킹 현상이 일어날 수 있으므로 바람직하지 않다. 반면에 출력 버퍼 방식의 경우, 즉 재결합 버퍼가 출력단에 위치하는 경우에는 스위치가 패킷에 속한 모든 셀들을 모아서 보낼 필요가 없으며 정상적인 ATM 스위칭과 마찬가지로 각각의 셀을 따로 보낼 수 있다. 따라서 재결합 버퍼를 출력단에 두는 것이 보다 효과적이며 따라서 본 논문에서는 출력 버퍼 방식의 구조를 고려한다.

본 논문에서 고려하는 VC 머징이 가능한 ATM LSR은 <그림 2>와 같은 non-blocking 스위치와 출력 모듈(OM, Output Module)을 가진다. 이 때 출력 모듈은 <그림 3>과 같이 여러 개의 재결합 버퍼, 출력 버퍼로 구성된다. 그림에는 표시하지 않았으나 출력 모듈에는 VCI/VPI 전환 모듈도 포함된다.

3. 체증 제어 알고리듬

3.1 체증 제어와 APD 알고리듬

비실시간 트래픽인 인터넷 트래픽은 TCP와 같이 상위 계층에서 에러를 검출하여 재전송하므로 ATM 계층에서 셀 손실이 발생하더라도 신뢰성있는 전송을 보장한다. 따라서 체증이 예상되는 경우 고의로 패킷을 폐기함으로써 흐름을 제어하여 상위 계층의 성능인 goodput을 저하시키지 않는 트래픽 제어 방법이 요구된다.

버퍼 고갈 상태에서 새로운 패킷이 입력되면 그만큼의 패킷을 폐기하여야 한다. 신뢰성있는 전송이 보장되는 인터넷 트래픽의 경우 이러한 패킷의 폐기에 의하여 체증 제어를 수행하게 되는데, 새로 입력되는 패킷을 폐기하는 방법인 TD(Tail Drop)가 대표적이다. 이 방법은 먼저 도착한 패킷을 먼저 전송하겠다는 서비스 원칙을 따르므로 이후 FIFO(First-In First-Out)로 표기하기로 한다. 이 방법은 이해하기가 쉽고 구현이 용이하다는 장점이 있지만, 현재 버퍼에 가득찬 패킷이 모두 서비스 받은 후에 상위 계층인 TCP에서 패킷 폐기를 인지할 수 있으므로 흐름 제어가 그만큼 지연되는 문제점이 있다(Nam, 1999).

이러한 기존 방법의 문제점을 개선하고 효과적인 체증 제어를 위하여 APD 알고리듬은 대략 두 가지 사항에 초점을 맞추어 설계하였다. 첫 번째 목적은 평균 큐 길이를 조절하여 체증 현상을 제어하는 것이고 두 번째 목적은 global synchronization을 피하는 것이다.

APD에서는 평균 큐 길이를 이용하여 초기 체증을 감지하고 평균 큐 길이와 현재 큐 길이를 이용한 이중 지수평활법을 사

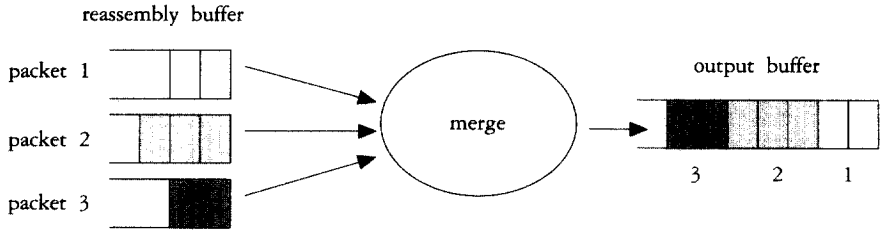


그림 3. 출력 모듈.

용하여 체증 제어를 도모한다. 만일 심각한 수준의 체증이 발생하면 도착하는 모든 패킷은 폐기되고 폐기된 패킷을 전송한 송신 호스트들은 slow start 단계에 들어가게 된다. 다수의 송신 호스트들이 거의 동시에 slow start 단계에 들어가게 되면 일시적으로 링크 사용률이 떨어지는 global synchronization이 발생한다. 이러한 문제를 해결하기 위해서는 망 종단 송신 호스트의 전송율을 조절하기 위한 소스 기반(source-based)의 체증 제어 방안이 필요하다. APD 알고리듬은 global synchronization에 의해 순간적으로 링크의 사용률이 극단적으로 감소하는 것을 방지하고 버퍼의 큐 길이를 항상 작은 크기로 유지할 수가 있도록 도모하고자 한다.

3.2 APD 알고리듬

<그림 4>에 APD 알고리듬에 대한 순서도를 도시하였다. APD에서는 버퍼의 평균 큐 길이(이후 Q_{avg}로 표기)를 이용하여 출력 모듈에 대하여 체증 제어를 수행한다. APD에서는, 체증통보(Congestion Notification)를 위한 ECN(Explicit Congestion Notification) 비트에 마크(mark)를 하거나, 패킷을 폐기(drop)하고, 송신 호스트는 해당 패킷의 처리여부에 따라 전송속도를 조절한다.

APD는 패킷을 마크나 폐기하는 척도로 매 패킷이 도착할 때마다 Q_{avg}를 구하고, 이것을 미리 정해놓은 파라미터인 최소 큐 한계값(minimum threshold, 이후 $\min_{th}$로 표기)과 최대 큐 한계값(maximum threshold, 이후 $\max_{th}$ 로 표기)과 비교한다.

Q_{avg}가 $\min_{th}$보다 작을 때에는 링크의 사용이 일정 수준보다 낮은 수준에 머물러 있다고 판단할 수 있으므로 모든 패킷은 정상적으로 처리된다.

Q_{avg}가 $\min_{th}$와 $\max_{th}$ 사이에 있을 때 도착하는 패킷은 확률 p_a에 의해 랜덤하게 폐기되든지, ENC 비트를 마크한다. <그림 5>에서와 같이 패킷 폐기 확률은 0에서부터 APD내에 정의된 상수인 최대 확률(maximum probability, 이하 $\max_p$로 표기)까지의 범위 내에 존재한다.

또한 Q_{avg}가 $\max_{th}$보다 클 때에는 Q_{avg}가 $\max_{th}$ 밑으로 떨어질 때까지 도착하는 모든 패킷을 폐기하거나 ENC 비트를 마크한다. 그러므로 Q_{avg}는 결코 $\max_{th}$를 벗어나지 못한다.

Q_{avg}는 이중 지수평활법을 통한 low-pass filter를 사용하여 다음과 같은 단계를 통해 계산한다.

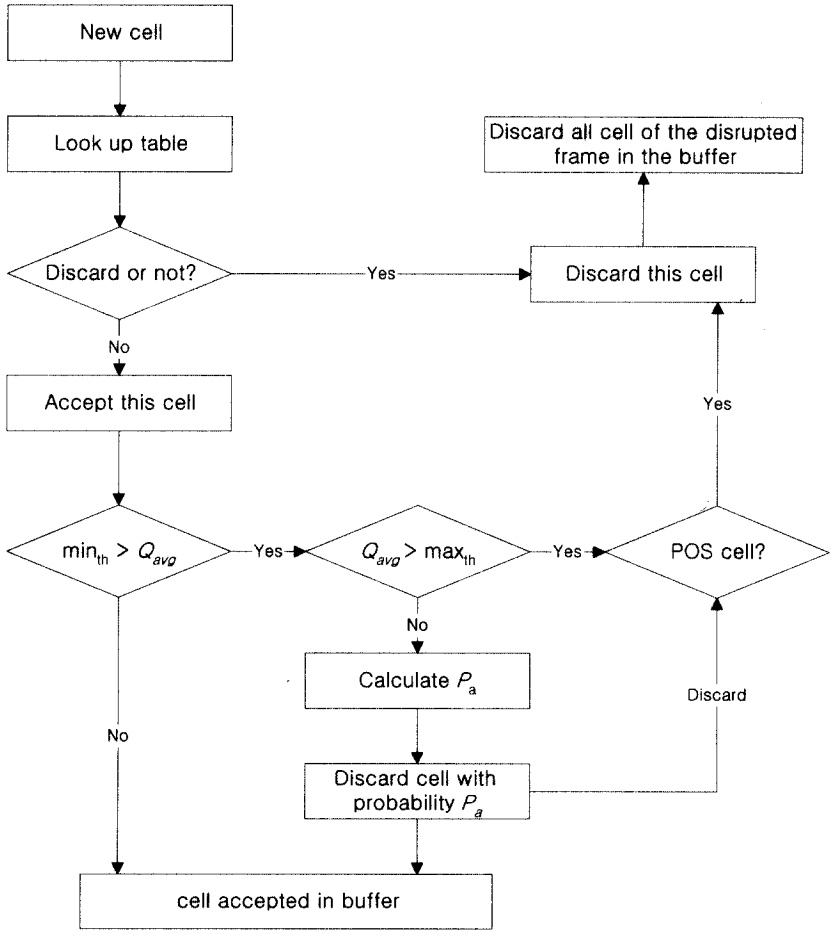


그림 4. APD 알고리듬의 순서도.

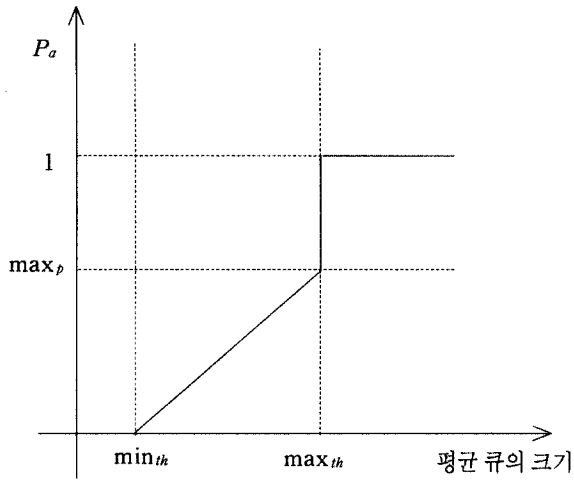


그림 5. 평균 큐의 크기에 따른 패킷 폐기 확률.

단계 1. 시간 t에서의 Q의 단순 지수평활법에 의한 큐의 길이(Y_t')를 계산한다.

$$Y_t' = (1 - w_q) \times Y_{t-1}' + w_q \times Q$$

w_q : 큐의 가중치(Queue Weight)

단계 2. 다음과 같이 Y_t를 계산한다.

$$Y_t = (Y_t' - Y_{t-1}')$$

단계 3. 시간 t의 지수평활추세(Q_t)를 다음과 같이 계산한다.

$$Q_t = Q_{t-1} + w_s \times (Y_t - Q_{t-1})$$

단계 4. Q_{avg}를 w_s를 적용하여 계산한다.

$$Q_{avg} = Y_t' + \frac{1 - w_s}{w_s} \times Q_t$$

w_s : 추세평활계수(Trend Smoothing Factor)

Q_{avg}를 구할 때 현재의 큐 길이가 Q_{avg}에 미치는 영향은 w_q와 w_s로 결정된다.

APD에서 패킷마크 여부를 결정하는 패킷마크확률(이하 p_a로 표기)은 Q_{avg}에 관한 함수로 다음과 같이 계산된다. Q_{avg}가 min_{th}에서 max_{th}까지 변화함에 따라 p_b는 선형적으로 0에서 max_p까지 값을 가질 수 있다.

$$p_b \leftarrow \frac{Q_{avg} - \min_{th}}{\max_{th} - \min_{th}} \times \max_p$$

$$p_a \leftarrow \frac{P_b}{1 - (count \times p_b)}$$

이와 같이 APD는 평균 큐 길이(Q_{avg})와 현재의 큐 길이를 이용하여 체증 제어를 수행하며 변화하는 트래픽을 파악, 제어하기 위하여 이중 지수평활법(double-EWMA)을 사용한다.

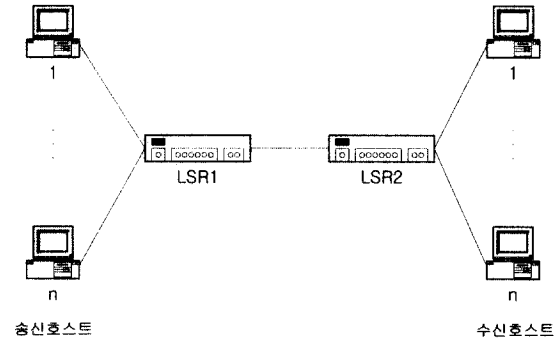


그림 6. 시뮬레이션 환경.

4. 성능 평가

4.1 시뮬레이션 구성

<그림 6>과 같은 네트워크 환경을 가정하여 시뮬레이션을 수행하여 APD 알고리듬의 성능을 기존의 알고리듬(이하 FIFO로 표기)과 비교하였다. 흐름 제어를 위해 각 호스트들은 2 Kbytes의 패킷을 congestion window size(cwnd)로 계속 전송하도록 하였다. 이에 따라 각 호스트들은 자신이 전송한 패킷이 체증되어 마크되거나 폐기되는 경우에는 slow start와 체증 회피(congestion avoidance)가 발생한다. 이때 호스트의 최대 윈도우 크기(advertised window size)는 모두 32 Kbytes로 하였고 타임아웃시간은 10.0 ms로 설정하였다(Yoo, Cho and Oh, 2001).

<그림 6>에서 송신호스트에서 LSR1까지의 전송지연시간은 2.0 ms, LSR1에서 수신호스트까지의 전송지연시간은 3.0 ms로 가정하였다. APD에 대한 파라미터의 경우, w_q값은 0.002, w_s값은 0.01, max_p는 0.02로 가정하였다. 또한 LSR의 버퍼 크기는 16패킷, 최소 한계값(min_{th})을 5패킷, 최대 한계값(max_{th})을 10패킷으로 설정하였다.

4.2 시뮬레이션 결과

<그림 7>, <그림 8>은 호스트가 4개일 때 VC 머징이 가능한 ATM LSR의 체증 제어에 대한 시뮬레이션 결과를 FIFO와 APD 알고리듬을 적용한 경우에 대하여 나타낸 것이다. 그림에서 알 수 있듯이 APD 알고리듬을 적용한 경우 버퍼에 대기중인 패킷의 수가 FIFO에 비하여 전반적으로 적은 것을 알 수 있는데, 이는 패킷의 변화에 민감하게 반응하여 체증 현상이 발생하지 않도록 제어하기 때문이다.

LSR에 연결된 호스트 수의 변화에 따른 LSR의 평균 큐 길이에 대한 시뮬레이션 결과를 <그림 9>에 나타내었다. 호스트 수가 증가하면 호스트당 패킷 수는 줄어들지만 LSR에 부가되는 총 트래픽은 증가하게 되는데, 동일한 트래픽하에서는 FIFO보다 APD의 평균 큐 길이가 작게 나타남을 알 수 있다. 이

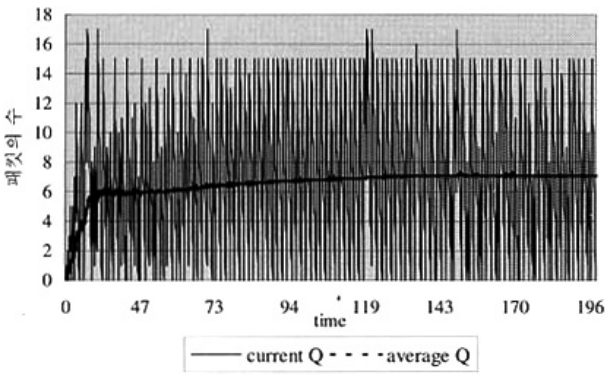


그림 7. 시뮬레이션 결과 (FIFO).

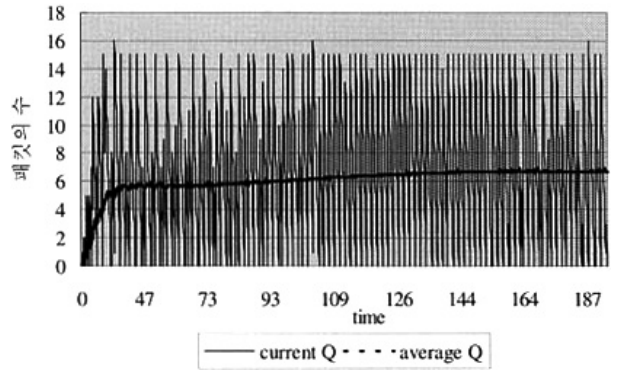


그림 8. 시뮬레이션 결과 (APD).

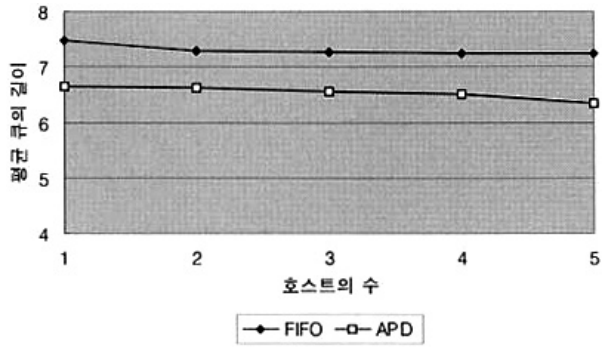


그림 9. 호스트 수의 증가에 따른 평균 큐 길이.

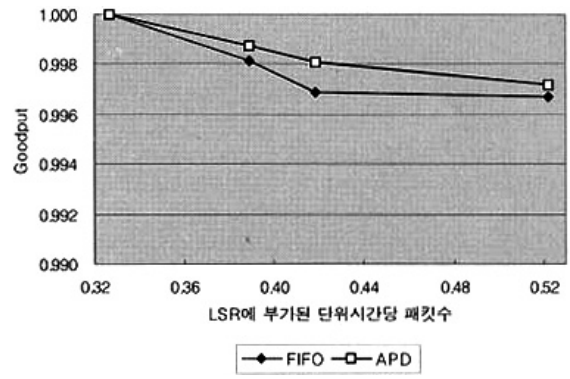


그림 10. 트래픽의 변화에 따른 goodput.

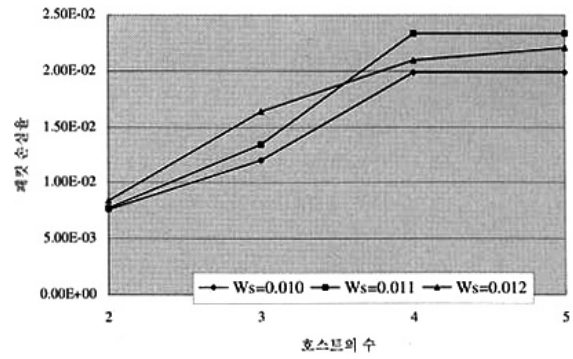


그림 11. 호스트 수에 따른 패킷 손실률.

는 APD가 FIFO에 비하여 총 트래픽의 증가에 능동적으로 반응하여 미리 패킷을 차단하기 때문이다.

트래픽의 변화에 따른 goodput을 <그림 10>에 나타내었다. 여기서 goodput은 부가되는 패킷 중에서 마크되지 않은 패킷(unmarked packet)의 비율을 의미한다. 그림에서 보면 트래픽이 증가하면 goodput이 감소하지만 전체적으로 FIFO보다는 APD가 높은 goodput을 가진다는 것을 알 수 있다. <그림 9>와 <그림 10>으로부터 대체적으로 APD 알고리듬이 FIFO에 비하여 우수한 성능을 가짐을 알 수 있다.

<그림 11>은 APD 알고리듬을 이용하는 경우에 대하여, LSR에 연결된 호스트 수의 변화에 따른 평균 패킷 손실률을 나타낸 것이다. 그림으로부터 호스트 수가 증가하여 LSR에 부가되는 트래픽 양이 증가하면 패킷의 손실이 전반적으로 증가하지만 부가되는 트래픽이 매우 커지더라도 일정한 수준의 손실률을 유지하는 것을 알 수 있다. 이는 APD 알고리듬이 총 트래픽의 증가에 능동적으로 반응하여 패킷 전송을 호스트 단위에서 제어하기 때문으로 체증 제어를 효과적으로 수행하고 있음을 알 수 있다.

5. 결 론

본 논문에서는 VC 머징이 가능한 ATM LSR의 구조에 대하여 기술하고 ATM LSR의 인터넷 트래픽에 대한 체증 제어 방법으로 APD(Active Packet Discard) 알고리듬을 제안하였다. 또한 시뮬레이션을 이용하여 APD 알고리듬에 대한 성능평가를 수행하였다. 시뮬레이션 결과에 따르면 본 논문에서 제안한 APD 알고리듬을 사용할 경우 기존의 운영방식인 FIFO에 비하여 트래픽의 증가에 능동적으로 반응함으로써 전반적인 성능이 우수하게 나타남을 알 수 있었다.

본 논문에서는 파라미터 값을 일반적인 경우로 가정하여 시뮬레이션을 수행하였는데, APD 파라미터 값을 동적으로 변화시키면 최적의 성능을 얻을 수 있을 것으로 생각되며 이는 추후 연구할 계획이다.

참고문헌

Callon, R., Swallow, G., Feldman, N., Viswanathan, A., Doolan, P. and Fredette, A. (1999), A Framework for MPLS, *IETF draft*.

Callon R. *et. al.* (1999), Multiprotocol Label Switching Architecture, *IETF draft*.

Chow H. K., Leon-Garcia A. (1999), VC-Merge Capable Scheduler Design, *IEEE ATM'99*, 153-160.

Iliadis, I. and Droz, P. (1996), Impact of VC Merging on Buffer Requirements in ATM Networks, *IFIP*.

Keith McCloghrie *et. al.* (1999), MPLS using ATM VC Switching, *IETF draft*.

Mario Baldi, Davide Bergamasco, Silvano Gai and Dante Malagrin (1999), CLIMAX: Cell-Interleaved Merged ATM conneXion, *Telecommunication Systems*, 12(1), 39-49.

Nam, H. (1999), Buffer Management for non-real traffic in ATM Switching Systems, *ETRI TM199902213*.

Yoo, J. W., Cho, Y. H. and Oh, Y. H. (2001), A Study on Reducing Buffer for VC-Merge Capable ATM Switch, *KICS*, 26(6A), 1060-1066.

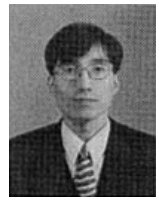
Widjaja, I. and Elwalid, A. I. (1999), Performance Issues in VC-Merge Cable switches for Multiprotocol Label Switching, *IEEE Communication*, 17(6), 1178-1189.

정 호 연

전남대학교 산업공학과 학사
서울대학교 산업공학과 석사
서울대학교 산업공학과 박사
현재: 전주대학교 산업공학과 부교수, 전주대학교 창업보육센터 소장, 전주대학교 기술혁신센터(TIC) 부원장
관심분야: 네트워크최적화, 유전알고리듬, 시스템 성능평가

서 재 영

전북대학교 공업화학과 학사
전북대학교 산업공학과 석사
현재: 전북대학교 산업시스템공학과 박사과정
관심분야: 통신시스템 모델링/성능분석



백 장 현

서울대학교 산업공학과 학사
서울대학교 산업공학과 석사
서울대학교 산업공학과 박사
현재: 전북대학교 산업시스템공학과 조교수
한국전자통신연구원 초빙연구원
관심분야: 통신망 설계, 트래픽 제어, 시스템 성능평가