

패킷 폐기 정책에 관한 연구

류동렬, 육동철, 박승섭
부경대학교 대학원 전자계산학과

A Study On Packet Dropping Policies

Dong-Ryeol Ryu, Dong-Cheol Yuk, Seung-Seob Park
Dept. of Computer Science, Graduate school, Pukyong Nat'l University

요 약

TCP 프로토콜과 ATM 프로토콜과의 형태 및 제어방식의 차이로 인한 이질성 문제 때문에 ATM층에서 한 개의 셀 손실은 상위 TCP층에서 하나의 패킷 손실을 초래하게 되어 망에 대한 충분한 QoS를 제공하지 못한다. 이러한 문제들을 해결하기 위해 처리율 면에서 Early Packet Discard(EPD)와 공정성 면에서 Selective Packet Discard(SPD), Fair Buffer Allocation (FBA) 등이 제안되었다.

본 논문에서는 ATM상에서 TCP 성능 향상을 위해 기존의 알고리즘과 제안한 알고리즘에 대하여 비교 분석하고, VC의 수를 가변적으로 변화시켜도 제안한 알고리즘은 SPD나 FBA 보다 나은 공정성을 보였고, 처리율 면에서는 차이가 없음을 시뮬레이션결과를 통해 나타내었다.

1. 서론

고속의 광 전송매체를 기반으로 한 ATM은 멀티미디어 통신을 가능하게 하는 고속 광대역종합통신망(B-ISDN)을 실현하기 위한 기본 방식으로 탄생된 것으로 멀티미디어 서비스에 다양한 QoS를 제공해야 한다. 인터넷의 TCP 서비스는 연결 지향 프로토콜로서 대중적으로 극적인 성장을 보였으며, 이질적인 네트워크뿐만 아니라 마이크로 컴퓨터, 미니 컴퓨터 등 거의 모든 컴퓨팅 시스템에서 사용되고 있다.

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.

이러한 문제들을 해결하기 위해 Early Packet Discard(EPD)[1][2][3][4]는 오버플로우 전에 AAL5 프레임의 첫 번째 셀이 어떤 임계치를 넘을 때 셀 전체를 폐기한다. EPD는 UBR상에서 TCP 데이터 처리 면에서는 매우 향상되었으나 공정성 면에서는 각 데

이터 전송을 하는 송신원 간 할당 대역폭 불균형이 발생된다. 이러한 대역폭할당에 있어 사용자들이 같은 통신 사용료를 내고 불공정한 대역폭 할당을 받는다 면 서비스의 신뢰성과 질의 저하를 가져온다.

EPD의 공정성 향상을 위해 SPD[5], FBA[6]가 서로간의 장점을 살려 공정성 향상을 위해 제시되었다.

본 논문은 SPD, FBA 알고리즘을 같은 네트워크 시뮬레이션 조건하에서 ATM상에서의 TCP 서비스 개선을 위해 공정성과 처리율을 비교 분석하였다. 본 논문 구성은 서론에 이어, 2장에서는 TCP 흐름제어와 폭주제어 방식, EPD, SPD, FBA 알고리즘에 관련한 설명과 3장에서는 공정성 개선을 위해 제안한 selective EPD with weight의 새로운 파라미터와 동작 방법에 관한 내용과, 4장과 5장에서는 시뮬레이션 환경 설정과 결과분석을 나타내었고, 마지막으로 6장에서는 결론에 대해 서술하였다.

2. 관련연구

2.1 TCP 흐름제어와 폭주제어

현재의 TCP에서 사용되는 기술은 다음과 같다

2.1.1 Slow Start 방식

Slow Start는 폭주 윈도우(congestion window)를 사용한다. 새로운 연결이 설정되었을 때 TCP 개체는 폭주 윈도우를 세그먼트크기와 같은 1로 초기화하고,

수신지에서 ACK가 수신될 때마다 폭주 윈도우를 최대값까지 1씩 증가시킨다. 따라서 ACK값이 지수적으로 증가함으로, 실제로는 폭주 윈도우도 지수적으로 증가한다.

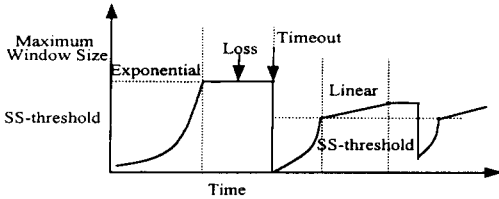


그림 1. Slow start와 congestion avoidance 단계의 폭주 윈도우의 크기 변화

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

2.1.2 Fast retransmission과 Fast recovery 방식

Fast retransmit은 TCP 버전(예: Reno, New Reno, Tahoe)에 따라 서로 약간의 차이는 있지만 손실된 세그먼트를 빨리 복구한다는 의미에서는 차이는 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 재전송함으로써 타임아웃이 만료될 때까지 기다려야하는 시간을 줄이는 방법이다.

Fast recovery는 세그먼트의 손실 시 폭주윈도우의 1/2을 SS-threshold로 설정하고, 폭주윈도우의 크기를 1로 설정하고서 slow start 단계로 들어가는 방법이 보수적이라고 판단하여, 이러한 방법 대신 세그먼트의 손실 시 폭주윈도우 크기의 1/2를 현재 폭주윈도우의 크기로 설정하고, 곧바로 congestion avoidance 단계로 들어가게 함으로써 Slow start 단계를 제거하는 방법이다.

2.2 EPD 알고리즘

Romanow는 ATM 망에서 TCP 서비스를 사용해서 패킷 기반의 TCP의 성능을 분석하였다[2]. 그들의 성능 분석 결과, 폭주 발생으로 인한 셀 폐기는 성능

이 저하될 수 있음을 보였다. 이러한 낮은 처리율은 전송로 상에 파괴된 패킷 전송으로 말미암아 대역폭 낭비를 가져오기 때문이다. 이 문제를 해결하기 위해 그들이 제안한 Early Packet Discard는 ATM 망에서 TCP의 성능을 개선 시켰다.

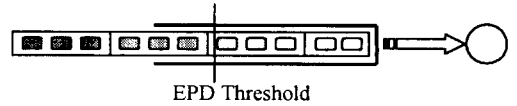


그림 2. EPD 방식과 EPD 임계치

그림 2와 같이 EPD 구현은 스위치의 버퍼에 임계치를 설정하는 것이다. 스위치는 큐가 주어진 임계치를 초과했을 때 들어오는 패킷들을 폐기한다. 여기서 임의로 패킷을 폐기하는것이 아니라 첫 번째 셀이 있으면 같은 가상 채널 식별자를 가진 셀을 EOM(End of Message)까지 전체 폐기한다. 만약 전체 버퍼가 꽉 찬 상태가 아닐 지라도, 첫 번째 셀이 폐기되지 않으며, 나머지 셀들도 폐기되지 않는다. 그리고 EPD는 PPD[6]와 많이 연결되어서 사용되어진다. 참고 논문 [2]에서는 EPD는 PPD와 DT보다 향상된 성능을 보였다. 그러나 EPD는 각 TCP 컨넥션간의 각 셀의 VCI에 대한 정보를 사용하지 않는 알고리즘이기 때문에 공정성에 대한 문제가 있다.

2.3 SPD 알고리즘

SPD는 EPD의 공정성을 개선하기 위해 참고 논문 [5]에서 소개되었다. 여러개의 TCP 컨넥션중에서 하나의 TCP 컨넥션이 패킷 전송을 한 후에, 수신측 ACK에 따라 Slow start 단계를 수행하기 위해 윈도우 크기를 증가시킨다. 그러나 다른 TCP 컨넥션은 버퍼 오버플로우에 의해 셀이 폐기되는 현상을 발생시킨다. 또는 다른 TCP 컨넥션은 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 수의 계산에 의한 활성화된 VC의 수에 대한 정보를 가진다. 여기서 VC 당 버퍼에서 적어도 하나의 셀을 가지다면, 활성화된 상태이다. 공정한 할당은 활성화된 VC의 수에 현재 버퍼의 큐길이를 바탕으로 계산된다.

그림 3에서 스위치에 들어온 TCP패킷을 VCj라 가정하고, 만약 VCj에 대하여 현재 버퍼크기가 $Q \geq Th$ 이거나 $Q_j \geq [Th]$ 인 경우 첫 번째 셀은 폐기된다. 여기서 Q_j 는 스위치 버퍼의 VCj에 해당하는 셀수이며, $[Th]$ 는 $= K * Q/N$ 으로 정의할 수 있으며 다시

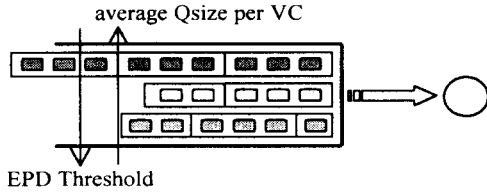


그림 3. SPD 방식

말해서 VC당 평균 큐 길이가 된다. N은 활성화된 VC의 수이고, K는 제어 파라미터(Control parameter)인데 전형적으로 $1 \leq K \leq 2$ 로 설정한다. SPD 알고리즘에서 Th는 $K = 1$ 일 경우 평균 큐 길이를 나타낸다.

2.4 FBA 알고리즘

스위치에 최대 연결된 VC의 수만큼 버퍼를 사용해서 활성화된 VC의 수에 현재 버퍼의 큐길이를 가지고 계산된다. Qj는 VCj의 큐 길이로 대응되며, 전체 큐 길이 Q가 $Q \geq Th$ 이고 $Q_j \geq [Th]$ 일 경우, VCj의 새로운 패킷은 폐기된다. [Th]는 다시 말해서 VC당 평균 큐 길이가 되며, $K \times Q/N$ 으로 정의한다. N은 활성화된 VC의 수를 의미하며, K는 제어 파라미터로 모든 VC의 큐는 라운드 로빈(Round Robin) 스케줄에 의해 처리된다. 이때 각 라운드 로빈에 의해 셀이 처리될 때, 오직 하나의 VC 큐 셀이 처리된다.

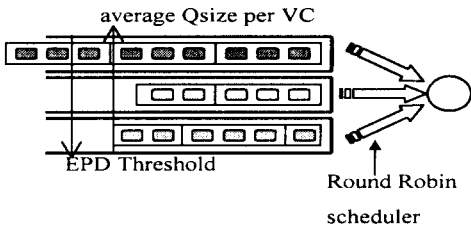


그림 4. FBA 방식

3. 공정성 개선을 위한 제안한 알고리즘 (selective EPD with weight algorithm)

앞장에서 설명한 알고리즘을 기반으로 본 연구는 그림 5와 같이 제안한 알고리즘의 의사코드를 나타낸다. 제안한 알고리즘의 중요한 내용은 임계치(Threshold) 처리에 가중치를 부여하여 선택적 셀 폐기 방식으로 버퍼에 셀들이 들어와서 폐기 또는 승인되는 정보를 임시 기억 장소에 저장함으로써, 임의의

VCj의 패킷이 얼마나 많이 폐기되고, 임의의 VCj가 다른 VCj+n 보다 얼마만큼의 패킷을 보냈는가를 알기 위해서이다.

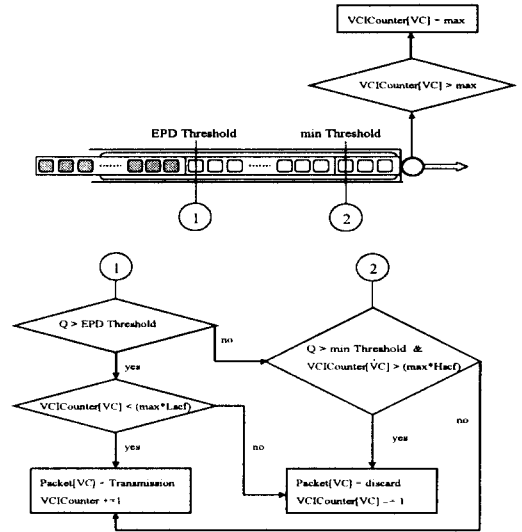


그림 5. 버퍼와 selective EPD with weight의 순서도

제시된 알고리즘에는 여러개의 파라미터가 추가되었다. 추가된 파라미터에 대한 설명은 표1과 같이 나타내었다. 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 계수를 증가시킨다. 이 때 스위치는 최고 많이 승인된 vciCount[VC]의 개수를 조사하여 max값을 갱신한다.

첫 번째 EPD Threshold에서 하는 역할을 기존 EPD에서의 역할과 유사하다. 단지 폐기를 할 때, low selective weight 파라미터를 사용하고, 버퍼에 패킷의 손실이 많은 VCj가 있을 경우에는 폐기를 하지 않고 패킷을 승인한다. 그리고 min Threshold는 승인된 패킷이 있을 지라도 high selective weight 파라미터를 사용해서 다른 VC 보다 패킷 승인 처리가 많을 경우에는 폐기를 시킨다. 추가된 파라미터인 min Threshold는 EPD Threshold - 1000값 만큼 설정하였고, high selective weight 값은 1/3, low selective weight

값은 2/3로 각각 설정하여 실험하였다.

표 1. 새로 추가된 파라미터에 대한 내용

새로 추가된 파라미터	내용
min Threshold	승인된 첫 번째 셀을 다시 검색하고, vciCount[VC]의 정보를 이용해서 승인 횟수가 $\max \times Hsw$ 보다 클 경우 승인이 아닌 폐기함으로써 각 VC 간의 공정성을 확보하기 위한 최소 임계치
max	패킷이 버퍼 처리가 승인될 때 vciCount[VC]에 계수가 올라간다. 여기서 최고 높은 계수를 가질 때, max 값 부여
high selective weight(Hsw)	vciCount[VC]는 승인된 패킷 갯수를 가진다. 여기서 최대 max를 기준으로, $\max \times Hsw$ 보다 큰 vciCount[VC]를 검색할 때 사용
low selective weight(Lsw)	최대 max를 기준으로, $\max \times Lsw$ 보다 작은 vciCount[VC]를 검색할 때 사용

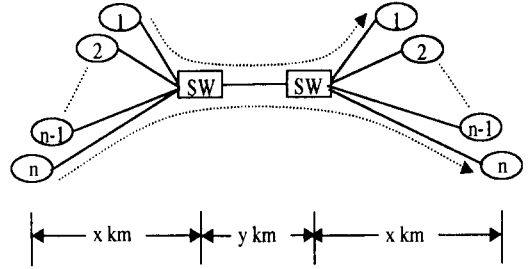


그림 6. 시뮬레이션 망 모델

4. 시뮬레이션 환경

본 장은 시뮬레이션 망 모델과 파라미터에 대해 설명한다.

4.1 망 모델

UBR서비스를 위한 여러 가지 알고리즘에서 TCP의 성능 분석을 위해서 그림 6과 같이 하나의 병목 구간을 가지는 peer to peer 모델로 설정하였다. 전송방향은 단 방향 트래픽(unidirectional traffic)으로만 데이터를 전송하게 하였다. 병목 구간의 링크 속도는 150Mbps로 거리는 300km로 가정하였다.

TCP 컨넥션의 수는 가변적으로 최대 n만큼의 수를 가진다. n은 최대 VC의 수가 최소 5에서 최대 70까지 범위를 설정하였으며, 각 TCP 컨넥션과 스위치 사이의 전송거리는 $n * 0.56$ km로 설정하였다.

그리고 송신원의 TCP 데이터는 고정 프레임 길이만큼 지속적으로 동시에 전송하게 하였으며, 시뮬레이션 시간은 총 3 초 동안 실험을 하였다.

4.2 시스템 파라미터 환경

각 UBR, TCP 컨넥션, UBR 스위치의 기본 파라미터 설정은 아래와 같이 설정하였다.

TCP Application

Data Frame = 512 bytes

TCP

Mean Packet Processing = 300 us

Send/Receiver buffer size = 64 Kbytes

Maximum segment size = 512 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

5. 결과 분석

성능 평가 요소로서는 각 VC 수에 따른 처리율과 공정성을 기준으로 하였다. 평가 요소인 공정성은 서로 다른 VC 간 대역폭 할당에 있어서 공평하게 할당 받았는가를 평가하는 기준으로 다음 식과 같이 주어진다[7][8]. 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, $O_i (i = 1 \dots n)$ 는 max-min의

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

최적의 처리율이고, n은 전체 VC의 수이다.

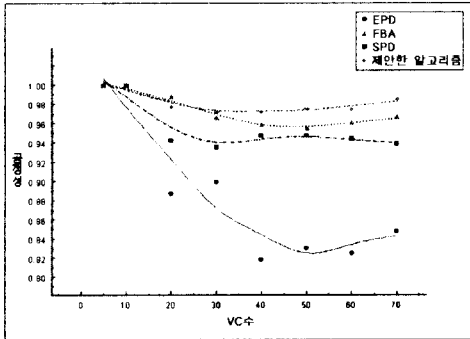


그림 7. 각 알고리즘의 공정성 비교

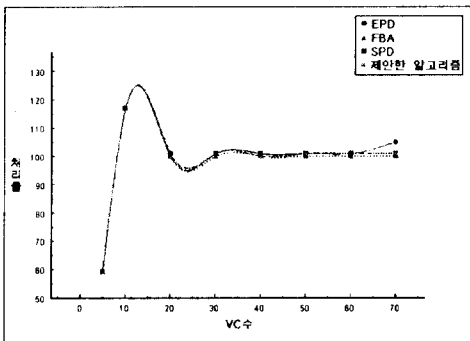


그림 8. 각 알고리즘의 처리율 비교

그림 7은 제안한 알고리즘과 EPD, SPD, FBA 알고리즘 간의 공정성에 대해 비교 분석한 그림이다. 기존 EPD의 공정성 문제를 SPD, FBA 알고리즘이 해결하였음을 알 수 있다. 그러나 본 연구에서 제시한 알고리즘은 기존의 알고리즘의 공정성 평가보다 나은 결과를 보이고 있다.

그림 8은 제안한 알고리즘과 기존의 알고리즘들 간의 처리율에 대한 결과를 나타낸 것이다. TCP의 세그먼트의 크기에 따라 차이가 나지만, 서로 같은 세그먼트의 크기로 기준으로 실험하여, 처리율에서는 EPD 알고리즘이 높고 제안한 알고리즘과 SPD, FBA는 차이가 없음을 알 수 있다.

6. 결론

본 논문에서는 ATM상에서 TCP 성능 향상을 위해 peer to peer망 모델에서 여러 가지 알고리즘과 제안

한 알고리즘에 대하여 비교 분석하였다. EPD의 공정성 문제해결을 위한 SPD와 FBA의 성능을 나타내었고, 제안한 알고리즘의 동작 알고리즘과 새로 추가된 파라미터에 대하여 설명하였다.

결론적으로 VC의 수를 가변적으로 변화시켜도 제안한 알고리즘은 SPD나 FBA 보다 나은 공정성과 처리율면에서도 차이가 없는 결과를 시뮬레이션을 통해 나타내었다.

[1] D. M. Drury, "ATM Traffic Management and the Impact of ATM Switch Design", Computer Networks and ISDN Systems 28, pp.471-479, 1996

[2] 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

[3] 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

[4] 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

[5] 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

[6] Heinanen J, Kilkki K, "A fair buffer allocation scheme", Computer Communications, V.21, p.220-226 Jan 1998

[7] L. Jaussi, M. Lorang, J. Nelissen, "A Detailed Experimental Performance Evaluation on TCP over UBR," IEEE ICATM '98, pp. 214-223, June 1998.

[8] 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