

ATM 망에서 효과적인 폴링 방법을 사용한 트래픽 정형화기의 설계

김형욱*, 최승원*, 조태경**, 최명렬*

한양대학교 제어계측공학과 ASIC & AI Lab, 425-791 경기도 안산시 사 1 동 1271*

동서울 대학 전기과, 461-714 경기도 수정구 북정동 423 번지**

TEL: +82-345-400-4036, FAX: +82-345-501-8114*, TEL: +82-342-720-2068**

{eternity, irick, choimy}@asic.hanyang.ac.kr*, tkcho@haksan.dsc.ac.kr**

A Fair Traffic Shaper Design Using a Efficient Polling Scheme in ATM Networks

Hyung-Wook Kim*, Seung-Won Choe*, Tae-Kyung Cho**, Myung-Ryul Choi*

Dept. of Control & Instrumentation Eng., Hanyang University

1271, Sa 1-Dong Ansan-Si KyunGi-Do, Korea, 425-791*

Department of Electric Engineering, Dong Seoul College**

TEL: +82-345-400-4036, FAX: +82-345-501-8114*, TEL: +82-342-720-2068**

{eternity, irick, choimy}@asic.hanyang.ac.kr*, tkcho@haksan.dsc.ac.kr**

요 약

본 논문에서는 CPU의 역할을 최대한으로 줄이고, 고속으로 트래픽을 정형화 하는 효율적인 정형화기를 설계하였다. 기존에 제시 되었던 대부분의 정형화 알고리즘들은 큐를 가지고 CPU에 의해 소프트웨어적으로 트래픽 정형화를 수행하게 된다. 본 논문에서는 주로 하드웨어적으로 정형화 작업을 수행하여 CPU의 부하를 줄여 고속으로 정형화를 수행할 수 있도록 BOP(Bandwidth Oriented Priority) Markovian 정형화기와 Circular 정형화기를 설계하고 시뮬레이션을 수행하였다.

1. 서 론

ATM(Asynchronous Transfer Mode)은 통계적 다중화(Statistical Multiplexing)를 통한 망 자원의 효율적인 이용 및 다양한 서비스를 수용할 수 있는 전송 방법으로, B-ISDN의 전송기술로 사용되어지고 있다. 멀티 서비스 망에서, 사용자는 망으로 전송할 자신의 데이터를 정확하게 표현할 수 있어야 한다. ATM 망에서는 자신의 데이터를 특정 것거나 표현하는 대신에, 표준이 되는 파라미터를 사용하여 자신의 트래픽 데이터를 표현한다[1].

망을 관리하는데 있어서 가장 핵심이 되는 것이 트래픽 관리(Traffic Management)이다. ATM 망은 셀 손실을 최소화 하여야 하기 때문에 망이 폭주상태에 빠지

기 전에 이를 방지 하는 것이 중요시 되고 있다. ITU-T의 권고안 I.371에서는 이러한 제어의 방법으로 호 수락 제어(CAC : Connection Admission Control), 사용자 변수 제어(UPC : Usage Parameter Control), 자원 관리(RM : Resource Management), 우선순위 제어(PC : Priority Control), 트래픽 정형화(Traffic Shaping) 등을 정의 하였다[1],[2]. 이 중에서 우리가 관심을 가지고 있는 트래픽 정형화는 망에 장애가 발생하지 않도록 출력단으로 출력되는 데이터 트래픽이 다음단에서 정해진 파라미터 값에 어긋나지 않도록 예방하는 일을 한다.

기존에 제시된 트래픽 정형화 알고리즘으로는 가장 간단하고 기본적인 Round Robin scheme이 있고, 그 외에 Cell-based schedulers, Connection-based multiplexer[3], WFQ[4], 시간차(Time Difference)를 이용한 정형화기[5] 등이 있다. Round Robin scheme을 제외한 대부분의 알고리즘들은 큐를 가지고 CPU에 의해서 소프트웨어적으로 트래픽 정형화를 행하게 된다. 그런데 정형화 기능을 모두 CPU에 의존하는 것은 CPU에 많은 연산 시간을 필요로 하여 망에서의 데이터 전송 속도가 떨어지게 된다. 이러한 이유로 대부분의 시스템에서는 정형화 기능이 있음에도 불구하고 실제로 이를 사용하지 않는 경우가 대부분이다. 따라서 CPU의 부하를 줄이면서 속도가 빠른 간단한 정형화기가 요구되어진다. 우리는 CPU의 역할을 최대한으로 줄이고, 고속으로 트래픽 정형화를 수행하기 위해 두가지 정형화기를 설계하고 시뮬레이션을 수행 하였다.

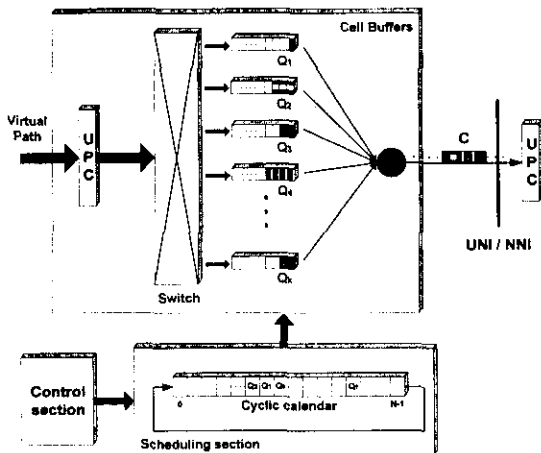
본 논문의 구성을 살펴보면, 2장에서 기존에 제안되

어 있던 정형화기의 종류와 특징을 살펴보고 3 장에서 두가지 정형화기를 제안하였다. 4 장에서는 제안된 정형화기에 대해 시뮬레이션을 수행 하였으며 5 장에서는 결론을 맺는다.

2. 정형화기의 종류와 특징

기존에 제안 되었던 많은 정형화 알고리즘들 중에서 Cell-based scheduler 와 Connection-based multiplexer 에 대하여 살펴 보겠다.

Cell-based 정형화 알고리즘은 다음 셀이 출력되는 시간을 매 셀이 도착 할 때마다 행하는 scheduler 를 사용하는데, scheduler 는 각 셀이 보내어질 때 다음 셀이 전송될 주기를 선택하는 일을 한다. 각 셀 주기에서 전송을 위한 다음 셀의 선택은 모든 큐에 걸쳐서 큐의 길이, 마지막 셀이 들어온 시간, 큐에서 머문 평균/최대 시간 등의 상태 파라미터들에 의해 정해진 방법에 따라 행한다. 그런데 이 알고리즘은 몇 가지의 문제점을 가지고 있다. 각 셀 주기에서 CPU 에 많은 연산 시간이 요구된다. 만약 메모리에 K 개의 큐를 가지고 있다면, 각 셀 주기에서 출력단으로 내보낼 셀을 선택하기 위해서는 적어도 K 번 비교 되어지게 된다. 그러므로 Cell-based scheduler 는 CPU 의 연산에 의한 지연 때문에 고속의 망(수백 Mbps ~ 수 Gbps)에는 적합하지 않다. 따라서 고가의 특별한 프로세서나 하드웨어가 사용된다. 그리고, 위의 시스템을 분석하면 CDV(Cell Delay Variation) tolerance 파라미터 τ 값을 정확하게 얻어낼 수 없다.



<그림 1> Calendar 를 기반으로 한 정형화기

Connection-based multiplexer 는 Cyclic calendar 를 사용하여 정형화를 수행하며, 오직 호 설정시에만 계산이

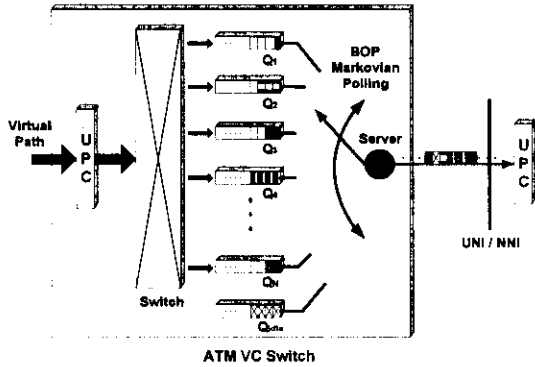
요구된다. 출력단으로 보내어지는 셀은 호 설정시 통계적으로 각 접속에 할당되고, multiplexing 은 접속이 완료될 때 까지 통계 된다. 호 설정을 위해서는 일반적으로 수 msec 정도의 시간이 요구되고, 효율적인 알고리즘은 이 시간을 슬롯을 할당 하는 문제를 해결 하는데 사용하게 된다. <그림 1>에서 볼 수 있듯이 Control section 에서 Cyclic calendar 를 만드는 것을 통제하는 구조로 되어있다. 즉, 호 접속시 Cyclic calendar 를 만들고 그 만들어진 것의 순서에 의해서 출력단으로 내보내어질 셀을 결정하게 되는 것이다. Calendar 를 만들 때는 전에 할당된 슬롯을 다시 재설정 하는 일이 없이 이루어 진다. 따라서, 출력 셀의 결정을 매 슬롯마다 계산하는 Cell-based 방법보다는 CPU 의 연산 시간을 줄일 수가 있다. 그렇지만, 이 방법은 접속이 늘어날수록, 각 접속의 대역폭이 커질수록 Calendar 의 크기가 계속 커져야만 하고, Calendar 를 만들기 위한 CPU 의 연산 시간이 역시 필요하다는 단점이 있다.

3. 제안된 트래픽 정형화기

셀은 입력단에서 UPC 를 거친 후 스위치를 거쳐서 N 개의 서로 다른 논리적인 큐를 가지고 있는 형태로 버퍼링하게 된다. 스위치로 셀들이 들어오기 전에는 UPC 를 통과하였으므로 각각의 QoS 파라미터에 맞도록 셀 간격이 정해져서 들어오게 된다. 하지만 스위치를 통과 하면서 해당 큐로 셀이 쌓여 들어 갈 때 셀의 간격이 협의에 의해 정해져 있던 간격보다 짧아질 수가 있다. 이렇게 큐에 쌓여 들어간 셀을 출력단으로 내보낼 때 정형화를 수행하지 않으면 다음 단의 UPC 를 거칠 때 정해진 파라미터를 준수하지 못하게 되어 셀을 태깅(Tagging)하여 다시 한번 기회를 주거나 폐기될 수가 있다. 따라서 정형화기에서 MUX 는 QoS 파라미터 값에 맞도록 셀을 다시 배치하여 출력단으로 내보내는 역할을 수행해야 한다. MUX 는 모든 VC 레벨의 접속들을 풀링하여 주는 기능을 수행하게 된다. MUX 가 풀링을 얼마나 빠른 시간에 효율적으로 QoS 파라미터를 준수하면서 행할 수 있는가가 정형화기의 성능을 좌우한다고 할 수 있다.

본 논문에서는 풀링의 방법으로 복잡한 알고리즘을 사용하는 대신, 주로 하드웨어적으로 풀링을 수행할 수 있도록 큐가 셀을 내보낸 후에 각 접속의 QoS 파라미터 값에 맞도록 다음 셀의 전송을 차단(Disable) 하는 방법을 사용하였다. 우리는 MUX 가 셀 전송을 할 수 있는 큐들을 선택할 때, BOP(Bandwidth Oriented Priority) Markovian 풀링 방법을 사용하였다. <그림 2>에서 BOP Markovian 풀링을 사용한 트래픽 정형화기의 구조를 나타내었다. MUX 는 Enable 신호를 내보내고 있

는 큐만을 대상으로 BOP Markovian 폴링을 수행하게 된다. 만약 Enable 되어 있는 큐가 여러 개일 경우, 우리가 제안한 방법에서는 대역폭이 큰 큐일수록 폴링될 확률이 높게 된다. 만약 모든 큐가 Disable 신호를 내보내고 있는 경우 그 시간 슬롯에는 Idle 셀을 삽입하도록 하였다.



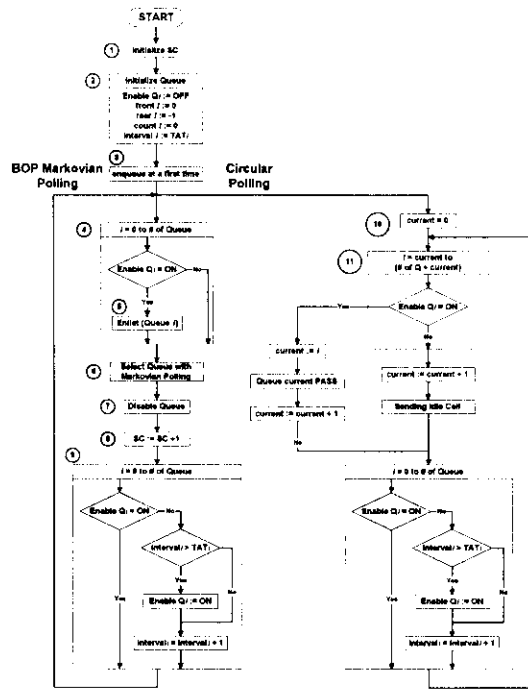
<그림 2> BOP Markovian 폴링을 사용한 트래픽 정형화기 구조

두 번째로 제안한 Circular 폴링 알고리즘도 BOP Markovian 폴링에서와 마찬가지로 큐의 상태를 Enable/Disable로 전환하면서 Enable 상태에 있는 큐들에 대해 행해진다. Circular 폴링은 Enable 신호를 내보내고 있는 큐를 대역폭에 관계없이 순서대로 순환적으로 폴링을 하게 된다. 즉, 순서대로 폴링을 해가다가 Disable 되어 있는 큐를 만나게 되면 그 시간 슬롯에서 바로 다음 큐의 Enable 여부를 판단하여 Enable 되어 있으면 폴링하고 Disable 이면 그 다음 큐를 다시 검사 하도록 하였다. 이미 Enable 신호를 내보내고 있는 큐는 QoS 파라미터에 맞도록 셀을 지연시킨 큐이므로 출력단으로 전송되는 셀은 모두 다음 단의 UPC를 통과할 수가 있게 된다. Circular 폴링 역시 모든 큐가 Disable 된 경우는 Idle 셀을 삽입한다. 그리고 Idle 셀이 삽입된 후에 다음 시간 슬롯에서는 이전에 폴링 했던 위치의 다음 큐부터 Enable 여부를 판단하게 된다.

<그림 3>의 순서도에서 ③번 이후에 좌측의 진행은 BOP Markovian 폴링을 사용한 정형화기의 흐름이고, 우측은 Circular 폴링을 했을 때의 흐름을 나타낸다. 전체적인 흐름은 다음과 같다.

- (1) SC (Slot Counter) : 0으로 초기화
- (2) Initialize Queue
- (3) 각 큐의 입력단에 셀이 들어왔는지 확인한 후에 넣는다.
- (4) 모든 큐를 검사
- (5) Enable $Q_i = ON$ 인 Queue $_i$ 를 list에 넣는다.

- (6) 5에서 list에 넣어진 큐들을 가지고 BOP Markovian 폴링 수행
- (7) 보내진 셀이 있는 큐는 Disable
- (8) SC 증가
- (9) 큐 관리 (Queue Management)
 - Disable 되어 있는 큐 중에서 TAT_i 이상 시간이 경과한 큐를 찾아서 Enable 시킴
- (10) 처음 폴링으로 선택된 큐 선택
- (11) current 변수에 해당하는 큐가 Enable/Disable 인지 검사



<그림 3> 두가지 제안된 알고리즘의 순서도

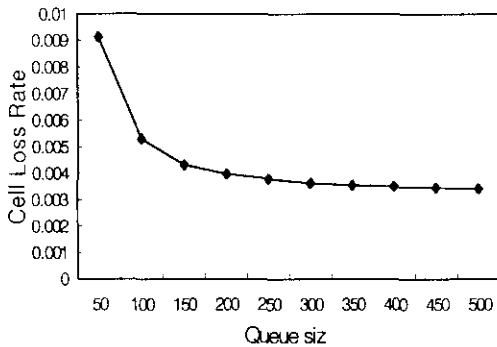
4. 시뮬레이션

우리는 물리적으로 155.52 Mbps 대역폭을 갖는 LAN(Local Area Network)의 데이터 트래픽이 유입된다고 가정하였고, VP로 대역을 나누고 그 대역 안을 VC로 다시 나누어서 사용하고 있다고 가정하였다. 155.52 Mbps의 대역폭을 가정하였으므로 한 슬롯 시간은 2.6 μ sec가 된다. 우리는 VC 레벨로 정형화를 수행한다고 가정하였으며, 한 개의 VP가 UPC를 통과한 후 ATM VC 스위치를 거쳐서 내부 테이블에 의해 VCI를 바꾼 후에 출력단으로 출력된다고 가정하였다.

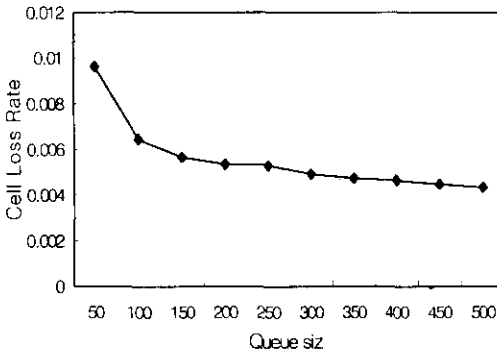
사용한 입력 트래픽 모델[6][7]로 우리는 지수 분포 (Exponential Distribution)를 사용하였다[8]. 셀이 버려지

는 경우는 각 큐로 입력되는 셀들이 큐를 넘칠 때 발생하며 메모리에 16 개의 논리적인 큐를 가지고 있는 구조를 가정하였다. 제안된 2 가지 정형화 방법을 각 큐의 크기를 변화 시켜가면서 시뮬레이션 하였다.

시뮬레이션 결과, 예상대로 큐의 크기를 증가 시킬 수록 셀 손실률이 줄어드는 것을 알 수가 있었다. <그림 4>는 BOP Markovian 풀링을 이용한 경우에 셀 손실률의 변화를 나타낸 것이고, <그림 5>는 Circular 풀링을 이용한 경우를 나타낸 것이다. 두 그래프에서 X 축의 단위는 각 큐의 개수이고, Y 축은 입력된 셀들의 총 개수와 Discard 된 셀의 개수의 비율 즉, 셀 손실률(CLR : Cell Loss Rate)을 나타낸다.



<그림 4> BOP Markovian 풀링을 사용한 정형화기의 셀손실률(Cell Loss Rate)의 변화



<그림 5> Circular 풀링을 사용한 정형화기의 셀손실률(Cell Loss Rate)의 변화

두 그래프의 단순 비교에서 나타나는 점은 BOP Markovian 풀링을 사용했을 때가 셀손실률이 적었다는 것이다. 이 점은 BOP Markovian 풀링이 확률적으로 대역폭이 큰 큐에 풀링을 더 많이 한다는 데서 예측되어

질 수 있다. 처음 큐 크기가 얼마 되지 않았을 때에는 큐 크기의 증가가 셀 손실률을 낮추는 효과를 크게 볼 수 있었다.

5. 결론

본 논문에서 제시된 알고리즘들은 한번 풀링한 큐는 QoS 파라미터 값에 의해 일정시간 큐로부터 출력되는 것을 막기 때문에 다른 정형화기들과는 달리 정형화 한 후의 모양이 전체 구간에 걸쳐서 매우 균등하게 이루어 지게 되는 것을 알 수 있다. 정형화 되어 출력되는 모양에서는 제안된 두가지 방식 모두 우수한 성능을 나타내나, 시뮬레이션 결과 BOP Markovian 풀링의 경우가 셀 손실률이 적었다. 우리의 알고리즘들은 기존의 정형화기들에 비해 간단하면서도 하드웨어로 구현 가능하도록 설계되었고, 제한된 큐의 크기를 가지고 상당한 성능을 발휘하는 것을 알 수 있었다. 앞으로의 과제는 CDV tolerance 파라미터 τ 값을 고려하여 정형화기를 설계하여 좀더 융통성을 주고, 하드웨어로 구현 하는 것이다.

참고문헌

- [1] ITU-T Recommendation I.371, "Traffic control and congestion control", June 1992.
- [2] ATM Forum, "Traffic Management Specification Version 4.0", March 1996.
- [3] Pierre-Andre Foviel, Aline Fichou, "Connection-based multiplexers: an alternative for traffic shaping in ATM networks", Proc. Of the IEEE ATM '97 Workshop, pp. 233 ~ 239, May 1997.
- [4] A. Parekh, R. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: the single node case, IEEE/ACM Trans. on Networking, Vol. 1, No. 3, pp. 344 ~ 357, June 1993.
- [5] Fang-Jang KUO, Jung-Shyr WU, "Design of Multi-Connection Shaper and Enforcer for Usage Parameter Control in ATM Networks", IEICE Trans. Commun., Vol. E79 B, No. 1, January 1996.
- [6] 임주환, 성단근, 한치문, 김영선, "ATM 교환", pp. 95 ~ 114, 1996.
- [7] Victor S. Frost, Benjamin Melamed, "Traffic Modeling for Telecommunications Networks", IEEE Communications Magazine, pp. 70 ~ 81, Mar 1994.
- [8] M. H. MacDougall, "Simulating Computer Systems Techniques and Tools", The MIT Press, 1987.