

자기 유사한 트래픽을 이용한 ATM 스위치의 ABR 혼잡제어 알고리즘의 성능분석

진 성 호[†] · 임 재 홍^{††}

요 약

네트워크를 설계하고 서비스를 구현하는데 있어서 중요한 변수중의 하나는 트래픽의 특성을 파악하는 것이다. 기존의 트래픽 예측과 분석은 포아송(Poisson) 또는 마코비안(Markovian)을 기본으로 하는 모델을 사용하였다. LAN, WAN 및 VBR(Variable Bit Rate) 트래픽 특성에 관한 최근의 실험적 연구들은 기존의 포아송 가정에 의한 모델들이 네트워크 트래픽의 장기간 의존성 및 자기 유사한 특성들을 과소평가 함으로써 실제 트래픽의 특성을 제대로 나타낼 수 없다는 것을 지적해 왔고 최근 실제 트래픽 모델과 유사한 모델로서 자기 유사한 특성을 이용한 접근법이 대두되고 있다. 따라서, 본 논문에서는 실제 트래픽과 유사한 자기 유사 데이터 트래픽을 백그라운드 부하로 발생시켜 기존의 ABR (Available Bit Rate) 혼잡제어 알고리즘 중 대표적인 EFCI(Explicit Forward Congestion Indication), EPRCA(Enhanced Proportional Rate Control Algorithm), ERICA(Explicit Rate Indication for Congestion Avoidance), NIST(National Institute of Standards and Technology) 스위치 알고리즘이 버스트한 트래픽에 대해 효율적으로 반응하는지를 ACR(Allowed Cell Rate), 버퍼 사용율, 셀 폐기율, 전송효율 등으로 나누어 분석을 하였다.

Performance Analysis for ABR Congestion Control Algorithm of ATM Switch using Self-Similar Traffic

Sung-Ho Jin[†] · Jae-Hong Yim^{††}

ABSTRACT

One of the most important matters in designing network and realizing service, is to grip on the traffic characteristics. Conventional traffic prediction and analysis used the models which based on the Poisson or Markovian. Recently, experimental research on the LAN, WAN and VBR traffic properties have been pointed out that they weren't able to display actual real traffic specificities because the models based on the Poisson assumption had been underestimated the long range dependency of network traffic and self-similar peculiarities, it has been lately presented that the new approach method using self-similarity characteristics as similar as the real traffic models. Therefore, in this paper, we generated self-similar data traffic like real traffic as background load. On the existing ABR congestion control algorithm, we analyzed by classify into ACR, buffer utilization, cell drop rate, transmission throughput with the representative EFCI, ERICA, EPRCA and NIST switch algorithm to show the efficient reaction about the burst traffic.

키워드 : ATM 스위치(ATM Switch), 자기유사 트래픽(Self-Similar Traffic), ABR 혼잡제어 알고리즘(ABR Congestion Algorithm)

1. 서 론

ATM(Asynchronous Transfer Mode) 망은 광대역 종합 정보통신망에서 요구하는 음성, 데이터, 영상 등 트래픽 특성과 성능이 서로 다르고 상이한 QoS(Quality Of Service)를 갖는 응용 서비스들의 모든 정보를 셀로 분할하여 통계적 다중화 방법에 의해 통일적이고 효율적으로 망을 통해 전송한다. 따라서 이런 다양한 QoS를 효율적으로 제공하기 위해 ATM 포럼에서는 각 서비스 계층을 트래픽 특성과

QoS 요구사항에 따라 다섯 개의 범주로 규정하고 있다.

트래픽의 비트율과 QoS에 따른 서비스 계층은 CBR(Constant Bit Rate), rt-VBR(real time Variable Bit Rate), nrt-VBR(non-real time Variable Bit Rate), UBR(Unspecified Bit Rate), ABR(Available Bit Rate)로 나뉘고 모든 서비스 계층은 동일한 물리 계층을 공유한다. CBR과 VBR 서비스 계층은 보장된 전송율과 대역폭을 사용하여 전송하고, UBR은 어떠한 대역폭이나 전송율을 보장받지 못하기 때문에 빈번한 셀 손실이 발생하므로 사용자 터미널간의 재전송 가능성이 가능한 어플리케이션에 유용한 서비스이다. ABR은 CBR이나 VBR이 사용하고 남은 대역폭을 사용하며 전송지연에 대해서는 민감하지 않으나 비교적 작은 셀 손실율을 요구하

[†] 준 회원 : 한국해양대학교 대학원 전자통신공학과

^{††} 정 회원 : 한국해양대학교 전자통신공학과 교수

논문접수 : 2002년 7월 24일, 심사완료 : 2002년 10월 1일

는 어플리케이션에 피드백 혼잡 제어 방법을 사용하여 전송 대역폭을 공평하게 배분하고자 하는 서비스 계층이다. 인터넷 데이터와 같은 매우 동적인 트래픽을 기존의 CBR, VBR로 전송하게 되면 링크 효율이 매우 낮아지는 문제점이 있어, 지연에 대해 엄격한 제한이 요구되지 않는 데이터 서비스의 경우 ABR 서비스를 이용하는 것이 효율적이다.

ATM 망에서 ABR 서비스를 위한 전송율 기반의 혼잡 제어 스위치는 크게 명시율 피드백(explicit rate feedback) 스위치와 이진 피드백(binary feedback) 스위치로 나눌 수 있다. 이진 피드백 스위치는 ATM 셀 헤더에 있는 EFCI 비트를 이용하여 망이 혼잡인지 아닌지를 구분하는 두 가지의 상태만을 수신원에 알리게 된다. 수신원은 도착하는 데이터 셀의 EFCI 비트를 조사하여 해당 VC(Virtual Connection)의 EFCI 상태를 테이블에 저장한 후, 해당 VC의 순방향 RM(Resource Management) 셀을 수신하면 테이블의 상태에 따라 역방향 RM 셀의 CI(Congestion Indication) 비트를 이용하여 송신원에 혼잡의 발생과 해제를 알리게 된다. 송신원에서는 전달되어온 혼잡 상태의 유무에 대한 정보에 따라서만 전송율의 증감을 결정하게 된다. 이에 반해 명시율 피드백 스위치는 VC 간의 공평한 몫인 공평한 공유(fair share)와 입력 부하의 정도에 따라 망에서 허용할 수 있는 전송율을 결정하여 이를 RM 셀의 ER(Explicit Rate) 필드에 실어 송신원에 알리고 송신원은 이 값을 이용하여 전송율을 조정하게 된다. 현재까지 ATM 포럼을 통하여 제안된 대표적인 ER 피드백 스위치의 예는 EPRCA 스위치와 혼잡 회피 방식을 사용하는 ERICA 스위치, 미국 NIST 연구소에서 개발한 NIST 스위치 알고리즘 등이 있다. 그러나 기존에 이러한 여러 가지의 스위치 알고리즘에 대한 성능분석의 연구 결과는 주로 트래픽 특성 모델링이 실제 운영중인 네트워크의 트래픽 특성과 많은 차이를 보인다는 것이다. 기존에는 포아송이나 마코비안을 기반으로 하는 큐잉 모델을 사용하였고 이러한 모델을 사용하였을 경우에는 실제 네트워크에서 발생하는 트래픽 특성과 많은 차이가 나고 있다. 현재의 트래픽 특성은 인터넷 사용자의 증가와 특히 트래픽의 종류가 일반적인 텍스트가 아닌 멀티미디어화로 인해 그리고 대용량의 파일 전송이 많아지면서 매우 버스트한 특성을 보이고 있는 반면, 기존의 모델을 사용한 경우 이러한 버스트한 특성을 나타내지 못하기 때문에 많은 문제점을 나타내고 있다. 최근에 이러한 문제점을 해결하기 위해 여러 연구가 진행되고 있으며 특히 자기 유사 특성에 관한 연구가 트래픽에 대한 모델로서 인증을 받고 있는 추세이다[1, 2]. 따라서 본 논문에서는 실제 트래픽과 유사한 자기 유사한 트래픽을 백그라운드 부하로 발생시켜 기존의 ABR 알고리즘 중 어느 알고리즘이 버스트한 트래픽에 효율적으로 반응을 하는지를 ACR, 버퍼 사용율, 셀 폐기율, 전송효율 등으로 나누어 분석을 하였다.

논문의 구성은 서론에 이어 2장에서 자기 유사한 트래픽의 특성에 대해 살펴보고, 실시간 운영중인 네트워크 트래

픽, 합성된 자기 유사한 트래픽 그리고 포아송 분포 특성을 가지는 트래픽을 비교 분석한다. 3장에서는 기존의 ABR 전송제어 알고리즘에 대해 살펴보고, 4장에서는 시뮬레이션 및 알고리즘에 따른 ACR, 버퍼 사용율, 셀 폐기율, 전송효율 등에 대해 비교 분석한 후 5장에서 결론을 맺는다.

2. 자기 유사 특성 및 트래픽 비교 분석

자기 유사성은 최근에 들어서 데이터통신 트래픽 분석에 적용된 중요한 개념이다. 자기 유사성은 차원상의 서로 다른 확률 대 비율이나 서로 다른 스케일에서 보았을 때 동일하게 보이거나 동일하게 행동하는 것이 자기 유사한 현상을 말한다. 즉, 데이터 트래픽의 시간 순서가 해상도의 정도에 상관없이 동일한 패턴으로 나타나는 것이다. 네트워크 성능의 관점에서 자기 유사한 특성은 집단화의 지속성이며, 포아송 트래픽에서 집단화는 단기간 내에 발생하지만 장기간이 지나면 부드러워진다[3].

일반적으로 자기 유사성은 결정론적인 시계열에 대해 구성되어진다. 그러나 데이터 트래픽은 확률적인 과정으로 가장 잘 고찰되어지고, 통계적인 방법으로는 자기 유사성을 언급할 수 있다. 일반적으로 결정적이고 주기적인 신호는 시간 이동에 관하여 불변인 특징이 있다. 즉 그 신호는 시간상으로 여러 주기가 이동하더라도 동일한 신호이다. 이에 반해, 정상확률 과정에 대해서는 그 과정의 통계는 시간 이동에 불변이다. 또한 평균은 시간에 독립적이고 자기상관 함수는 단지 시간의 차이에만 의존한다.

자기 유사성 확률과정은 기존의 논문에서 여러 가지 방법으로 정의되어져 왔다. 본 논문에서는 먼저 연속시간 확률 과정에 대해 살펴본 후 데이터 트래픽과 관련된 이산시간 확률과정에 대해 살펴본다[1, 4].

2.1 연속시간 정의

자기 유사성 확률과정의 일반적인 정의는 다음과 같이 연속시간 변수의 직접 스케일링에 기초한다. 어떠한 실수 $a > 0$ 에 대해, 확률과정 $a^{-H}x(at)$ 가 $x(t)$ 와 통계적으로 동일한 특성을 가진다면, 확률과정 $x(t)$ 는 파라미터 $H(0.5 \leq H \leq 1)$ 를 가지고 통계적으로 자기 유사하다. 이러한 관계는 식 (1)~식 (3)의 3가지 조건으로 표현된다.

$$E[x(t)] = E[x(at)] \quad \text{Mean} \quad (1)$$

$$\text{Var}[x(t)] = \frac{\text{Var}[x(at)]}{a^{2H}} \quad \text{Variance} \quad (2)$$

$$R_x(t, s) = \frac{R_x(at, as)}{a^{2H}} \quad \text{Autocorrelation} \quad (3)$$

Hurst 또는 자기 유사 파라미터 H 는 자기 유사성의 핵심척도이다. 다시 말하면, H 는 통계적인 현상의 지속성에 대한 척도이고 확률과정의 장기간 종속에 대한 척도이다. $H=0.5$ 의 값은 자기 유사성의 부재를 나타내고, H 가 1에

가까울수록 지속성의 정도 또는 장기간 종속의 정도는 더욱 커진다[3, 4].

2.2 이산시간 정의

정상 시계열(stationary time series) x 에 대해, m-aggregated 시계열 $x^{(m)} = \{x_k^{(m)}, k = 0, 1, 2, \dots\}$ 은 인접한 m 크기의 블록을 겹침없이(nonoverlapping) 원래의 시계열을 합계함으로써 정의한다. 이에 대한 정의는 식 (4)와 같이 표현된다.

$$x_k^{(m)} = \frac{1}{m} \sum_{i=km-(m-1)}^{km} x_i \quad (4)$$

만약 식 (4)의 확률과정의 통계(mean, variance, correlation 등)가 동일한 압축 사본을 간직하고 있다면, 자기 유사한 과정으로 다룰 수 있다. 확률과정 x 가 모든 $m = 1, 2, \dots$ 에 대해서 식 (5), 식 (6)과 같다면 파라미터 $\beta (0 < \beta < 1)$ 에 대하여 정확하게 자기 유사하다고 한다.

$$Var(x^{(m)}) = \frac{Var(x)}{m^\beta} \quad \text{Variance} \quad (5)$$

$$R_{x^{(m)}}(k) = R_x(k) \quad \text{Autocorrelation} \quad (6)$$

파라미터 β 는 앞에서 정의했던 Hurst 파라미터 H와 관련하여 $H = 1 - (\beta/2)$ 로 나타내어진다. 정상과정 및 에르고딕 과정(ergodic process)에서는 $\beta = 1$ 인 경우로써, 시간 평균의 분산은 $1/m$ 비율로 감소하게 되지만, 자기 유사 과정에서는 시간평균의 분산은 더욱 천천히($1/m^\beta$) 감소하게 된다. 그러므로 $H = 0.5$ 의 값은 자기 유사성의 부재를 나타내고, H가 1에 가까울수록 지속성의 정도 또는 장기간 종속의 정도는 더욱 커진다. 확률과정 x 가 충분히 큰 모든 k에 대해 식 (7), 식 (8)과 같다면 근사적으로 자기 유사하다고 한다.

$$Var(x^{(m)}) = \frac{Var(x)}{m^\beta} \quad \text{Variance} \quad (7)$$

$$R_{x^{(m)}}(k) \rightarrow R_x(k), \text{ as } m \rightarrow \infty \quad \text{Autocorrelation} \quad (8)$$

따라서, 이러한 자기 유사성의 정의에 의해서, 집합 과정의 자기상관은 원래의 확률과정과 동일한 형태를 가진다. 이것은 변이성 또는 버스트한 정도가 서로 다른 타임 스케일에서 동일하게 나타난다는 것을 의미한다[1-3].

2.3 트래픽 비교분석 및 Hurst 파라미터 측정

본 논문에서는 실제 네트워크에서 발생하는 트래픽이 어느 정도의 자기 유사한 특성을 가지고 있는지, 그리고 기존의 포아송 모델이 자기 유사한 트래픽 특성을 나타내지 못하고 있음을 나타내기 위해 포아송과 자기 유사한 트래픽을 발생시켰고 실시간으로 운영중인 네트워크에서 트래픽을 수집하였다.

실시간 운영중인 네트워크 트래픽 샘플을 측정하기 위해

사용한 네트워크 환경은 본 저자 소속기관의 네트워크를 이용하였다. 샘플 측정은 C급 클래스인 203.230.252 네트워크에서 측정을 하였고 샘플 측정 도구는 HP사의 Internet Advisor를 사용하였다. 측정 파라미터는 전체 패킷수를 측정하였으며, 패킷 수는 신뢰성을 기하기 위하여 100만개 이상을 1초 단위로 측정하였고 측정 기간은 2002년 6월 12일부터 6월 26일까지 측정을 하였다. 그리고 자기 유사한 트래픽 모델링을 위해 100개의 독립적인 Pareto-like ON/OFF 소스모델을 중첩시켜 발생시켰다. ON 기간에서는 단위크기의 패킷이 CBR로 발생하고, OFF 기간에서는 트래픽이 발생하지 않는다고 가정하였다. 또한 포아송 트래픽 모델링을 위해 동일한 모델을 적용하여 ON, OFF 주기를 포아송 분포를 사용하여 트래픽을 발생 시켰다.

자기 유사성을 나타내는 기준으로 많이 사용되는 Hurst 파라미터를 구하는 방법은 Variance-time plot, R/S plot, Periodogram 방법 등 여러 가지가 있으며 본 논문에서는 가장 간단한 Variance-time plot 방법을 사용했다. 각 트래픽에 대한 Hurst 파라미터는 <표 1>과 같다. 실제의 트래픽과 자기 유사한 트래픽 모델링에 의해 인위적으로 생성된 트래픽의 경우 1에 가까운 아주 높은 Hurst 파라미터 값을 가지고 있으며 이는 자기 유사성의 강도가 아주 강함을 나타낸다. 이에 반해, 기존의 포아송 과정으로 생성한 트래픽의 경우 0.5에 가까운 H 값을 가진다. 이러한 결과에서 알 수 있듯이, 자기 유사한 트래픽 모델링에 의해 발생된 트래픽이 실제 트래픽 특성을 잘 나타내는 반면, 포아송 과정의 경우 실제 트래픽의 버스트한 성질을 제대로 나타내지 못함을 알 수 있다[5-8].

<표 1> 각 트래픽에 대한 Hurst 파라미터 값

트래픽 종류	β	H
실시간 트래픽	0.1054	0.9473
자기 유사 트래픽	0.1688	0.9156
포아송 트래픽	0.9478	0.5261

3. ABR 트래픽 알고리즘

3.1 EFCI 스위치 알고리즘

EFCI 알고리즘은 데이터 셀의 헤더에 있는 EFCI 비트를 이용하여 망의 혼잡 여부를 구분하는 두 가지 상태만을 수신원에 알리게 된다. 스위치는 모든 VC에 대해 공통의 출력 큐를 두고 큐의 크기를 감시하여 큐의 크기가 임계치를 넘으면 혼잡 발생으로 인식하여 전송되는 데이터 셀의 EFCI 비트를 마킹하여 혼잡이 발생하였음을 수신원에 알린다. 수신원은 EFCI 비트가 설정된 데이터 셀을 받게되면 전송할 역방향 RM 셀의 CI 비트를 1로 설정함으로써 망에 혼잡이 발생하였음을 송신원에 알리며 전송율을 감소하도록 한다. 대부분의 EFCI 스위치는 하나의 임계치를 사용할 때 발생할 수 있는 빈번한 혼잡 통지와 해제를 방지하기

위해 두 개의 임계치를 도입하고 있다. 즉 혼잡 발생을 선언하는 임계치 T_{high} 와 혼잡해제를 결정하는 T_{low} 를 사용하여 버퍼의 수위가 T_{low} 를 지나 T_{high} 를 넘으면 혼잡을 선언하고 버퍼가 줄기 시작해도 T_{low} 보다 낮지 않으면 혼잡 상태를 해제하지 않는다. 이 방식은 모든 VC가 공통의 버퍼를 공유하기 때문에 VC 간의 불공평성이 발생할 수 있다. 불공평성 문제의 개선책으로는 각 VC에 분리된 버퍼를 제공하는 방법과 선택적인 피드백을 통한 지능적인 혼잡제어 방법이 있다[9, 10].

3.2 EPRCA 스위치 알고리즘

EPRCA는 Roberts에 의해 1994년 9월 ATM 포럼에 소개된 전송율 기반 혼잡제어 방식으로 기존의 PRCA(Proportional Rate Control Algorithm) 방법에 공평한 공유 계산을 통한 ER 개념을 처음 도입하여 VC 간의 불공평성을 크게 개선한 방식이다. EPRCA 스위치에서는 RM 셀의 CCR(Current Cell Rate) 값을 사용한 지수 가중 평균(exponential weighted averaging) 계산 방법을 이용하여 식 (9)와 같이 MACR(Mean Allowed Cell Rate)을 구하고, 구해진 MACR을 이용하여 식 (10)과 같이 공평한 공유를 계산한다.

$$MACR = (1 - AVF) \times MACR + AVF \times CCR \quad (9)$$

$$Fair\ Share = DPF \times MACR \quad (10)$$

여기서 AVF(Averaging Factor)는 지수 가중 평균치를 위한 계수로서 1/16이고 DPF(Down Pressure Factor)는 스위치의 상황에 따라 1보다 작은 값으로 설정된다. EPRCA 알고리즘은 공평한 공유 값의 계산을 위해 순방향 RM 셀의 CCR 필드 값을 사용하여 스위치가 현재 혼잡 상태이면 CCR이 MACR보다 낮은 VC에 대해 식 (9)를 사용하여 MACR을 갱신한 후 스위치가 혼잡 상태가 아니면 CCR이 $MACR \times 7/8$ 보다 큰 VC에 대해 역시 식 (9)를 사용하여 MACR을 갱신한다. 스위치는 식 (9)와 식 (10)과 같이 구해진 MACR을 이용하여 공평한 공유 값을 계산하고 필요에 따라 역방향 RM 셀의 ER 필드값을 공평한 공유 값에 근거하여 줄인다. 스위치는 경미한 혼잡 상태와 심각한 혼잡 상태를 나타내는 두 개의 임계치를 가지고 있다. 먼저 심각한 혼잡 상태이면 모든 VC에 대해 ER을 $1/4 \times MACR$ 이하로 감소시키고, 경미한 혼잡 상태이면 CCR이 MACR의 $7/8$ 보다 큰 VC에 대해서만 선택적으로 ER을 $15/16 \times MACR$ 이하로 감소시킨다[11].

3.3 ERICA 스위치 알고리즘

혼잡 회피 방식은 혼잡을 미연에 방지하기 위해 평형 상태에서 스위치를 약간 저 부하 상태로 제어하여 버퍼에 셀이 거의 쌓이지 않게 조절한다. 이를 위해 목표 이용율을 1보다 약간 낮은 값으로 설정하고, 정해진 측정기간마다 실측을 통하여 실제 입력율과 가용 대역폭에 목표 이용율을 고려한 목표 전송율의 비율인 부하 지수를 식 (11)과 같이

계산하여 ER 값의 계산에 이용하는 방식이다.

$$Load\ Factor = \frac{Input\ Rate}{Target\ Cell\ Rate} \quad (11)$$

ERICA 스위치 알고리즘은 원래 OSU(Ohio State University)의 Jain 등에 의해 제안된 전송율 기반 혼잡제어 방식의 하나인 OSU 스위치에서 발전한 혼잡 회피형 스위치 알고리즘이다. ERICA 스위치 알고리즘은 순방향 RM 셀을 전송 받았을 때 VC 단위로 CCR 값을 저장하고, 부하에 근거한 ER 값을 식 (12)와 같이 계산한다.

$$ER\ based\ on\ load = ERI = \frac{CCR}{Load\ Factor} \quad (12)$$

그리고 VC들간의 공평성을 보장하기 위해 식 (13)과 같이 공평한 공유를 계산하고, 공평한 공유 값과 스위치에 인가된 부하에 근거하여 계산된 ER을 함께 고려하여 식 (14)와 같이 ER 값을 계산한다.

$$Fair\ Share = \frac{Target\ Cell\ Rate}{Number\ of\ Active\ VCs} \quad (13)$$

$$ER = \max(fairshare, ERI) \quad (14)$$

ERICA 스위치 알고리즘은 혼잡의 회피성, 빠른 응답성, 파라미터 설정의 용이성 등의 특징을 만족하게 설계되었다. 이후 ERICA 스위치 알고리즘은 성능의 개선을 위하여 알고리즘이 대폭 수정되어 ERICA+ 알고리즘으로 제안되었다 [10, 12, 13].

3.4 NIST 스위치 알고리즘

NIST 스위치 알고리즘은 NIST의 Chang 등이 제안한 혼잡 회피 기술을 사용하는 스위치 알고리즘이다. NIST 스위치 알고리즘에서는 EPRCA 스위치 알고리즘과 같이 계산된 MACR을 사용하여 ER 값을 계산하지만, 계산된 부하 지수가 1보다 작을 경우에는 스위치를 저부하 상태로 판단하여 잉여 대역폭을 재할당한다. 즉, 식 (15)와 같이 MACR을 일정한 값만큼 증가시켜 링크 효율을 개선시켰다. 여기서 MAIR(MACR Additive Increase Rate)은 가산 증가율로 0.5 Mbps로 설정된다.

$$MACR = MACR + MAIR \quad (15)$$

그리고 큐 길이의 증감을 이용하여 빠르게 혼잡 상황을 파악할 수 있는 조기 혼잡 검출 방식을 사용하였다. 즉 스위치의 큐 길이가 임계치 Q_T 를 넘고 변화율이 양이면 혼잡 상태로 판단하고, 식 (16)과 같이 ER값을 결정한다.

$$ER = MACR \times MRF \quad (16)$$

여기서 MRF(Major Reduction Factor)는 혼잡 시에 MACR을 감소시키는데 사용되는 상수로서 일반적으로 0.95로 설

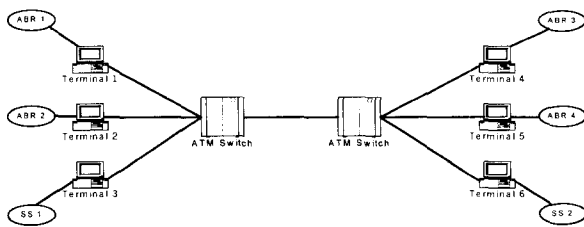
정된다. 그리고 스위치가 혼잡 상태가 아닐 경우는 ER 값은 식 (17)과 같이 설정된다[14].

$$ER = MACR \quad (17)$$

4. 시뮬레이션 및 결과 분석

4.1 시뮬레이션 모델

본 논문에서는 자기 유사한 트래픽 하에 여러 가지 ATM 혼잡제어 알고리즘을 평가하기 위해 (그림 1)과 같이 6개의 소스와 2개의 스위치를 갖는 시뮬레이션 모델을 사용하였다. 링크의 전송속도는 모두 155Mbps이고 스위치와 스위치의 거리는 1Km, 그리고 터미널과 스위치의 거리도 1Km로 단순한 모델을 설정하였다. 또한 본 논문에서는 자기 유사한 트래픽에 대한 ATM 혼잡제어 알고리즘의 반응성만을 분석하는데 초점을 두었기 때문에 공평한 공유에 대한 부분은 배제하였다.



(그림 1) 시뮬레이션 네트워크 구성

링크의 물리적인 지연과 스위치 내부의 동작 시간은 무시되었고, 스위치는 CBR, VBR, ABR, UBR 별로 트래픽의 종류에 대해 각각의 큐를 가지고 있다. 시뮬레이션에서 사용된 네트워크 모델은 ABR1, ABR2, SS1 (Self-Similar) 트래픽이 두 개의 ATM 스위치를 경유하여 ABR3, ABR4, SS2에 각각 전송되도록 하였다. ABR1, ABR2, SS1은 각각 50Mbps로 데이터를 전송하며 SS1은 Hurst 파라미터 H 값에 따라 버스트한 트래픽을 발생하여 스위치가 혼잡 상태로 빠졌을때 EFCI, EPRCA, ERICA, NIST 스위치 알고리즘의 반응을 살펴볼도록 구성되어 있다. 시뮬레이션에 사용된 송신원과 스위치의 ABR 파라미터는 ATM 포럼의 TM 규격을 따랐다. 본 논문에서 사용한 ABR 파라미터 값과 스위치 파라미터 값은 <표 2>, <표 3>과 같다.

<표 2> ABR 파라미터

Simulation Parameters	Values
Nrm (Number of RM cells)	32 cell
ACR (Allowed Cell Rate)	dynamic
ICR (Initial Cell Rate)	7.49 Mbps
PCR (Peak Cell Rate)	150 Mbps
MCR (Minimum Cell Rate)	1.49 Mbps
RIF (Rate Increase Factor)	0.0625
RDF (Rate Decrease Factor)	0.0625

<표 3> 스위치 파라미터

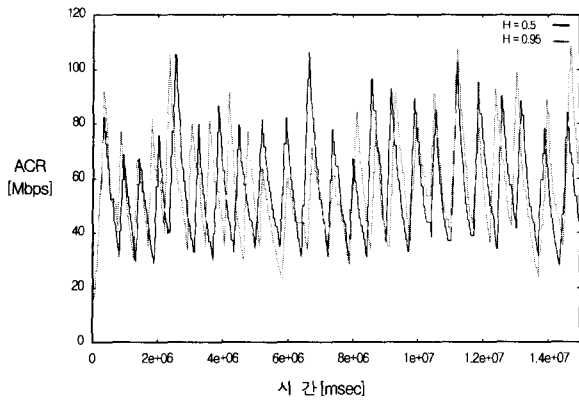
Simulation Parameters	Values
Queue size	100 cell
QHigh	50 cell
QLow	25 cell
Target Rate	142 Mbps
MAIR (MACR Additive Increase Rate)	0.5
MRF (Major Reduction Factor)	0.95
DPF (Down Pressure Factor)	0.875
Average Factor	0.0625

4.2 알고리즘에 따른 ABR 분석

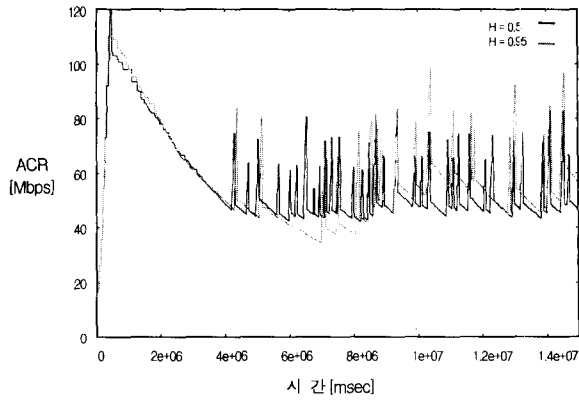
(그림 2)는 각 스위치 알고리즘에 대한 ACR의 변화를 나타낸 것이다. H 값의 변화에 따른 ACR의 변화를 살펴보면 (그림 2)(a)는 EFCI 스위치 알고리즘을 적용했을 경우로 ACR에는 큰 차이점이 없었다. EFCI 스위치 알고리즘은 단순히 스위치의 혼잡 상황으로만 ACR을 변화시키기 때문에 자기 유사한 트래픽의 버스트함에 ACR은 큰 영향을 받지 않음을 알 수가 있다.

(그림 2)(b)는 EPRCA 스위치 알고리즘을 적용했을 경우다. EPRCA 스위치 알고리즘은 심각한 혼잡 상태이면 모든 VC에 대해 ER을 $(1/4) \times MACR$ 이하로 감소시키고, 경미한 혼잡 상태이면 CCR이 MACR의 7/8보다 큰 VC에 대해서만 선택적으로 ER을 $(15/16) \times MACR$ 로 감소시킨다. 그러므로 버스트한 트래픽이 유입되었을 때 MACR이 급격하게 감소하게 되고, 이로 인해 경미한 혼잡 상태가 된 이후 혼잡이 사라질 때까지 MACR이 천천히 감소하게 되는 것을 알 수가 있다. H 값에 따라서는 H=0.95일 때가 H=0.5일 때보다 ACR의 진동율이 작다는 것을 알 수 있다. 이것은 H=0.95일 때는 H=0.5일 때보다 트래픽의 버스트 강도가 아주 심하기 때문에 스위치 큐의 셀 누적이 심해지고, 따라서 혼잡의 상황이 지속되기 때문이다.

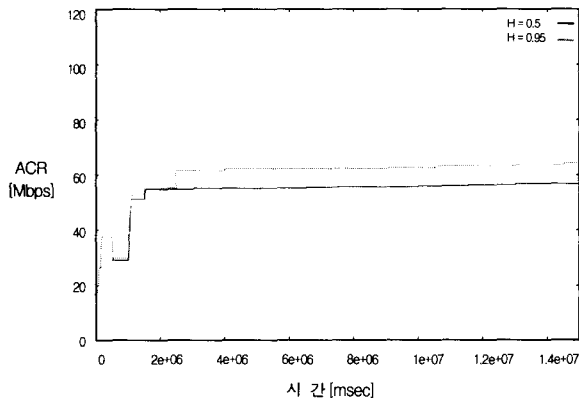
(그림 2)(c)는 ERICA 스위치 알고리즘을 적용했을 경우다. ERICA 스위치 알고리즘은 ER 값을 결정할 때 공평한 공유와 ERI중 최대의 값을 선정한다. 그러나 혼잡이 발생하여 ERI 값을 계산할 때 스위치 큐 상태를 파악하여 스위치의 혼잡을 파악하는 것이 아니라 VC 단위로 저장된 CCR 값과 부하지수를 이용하여 ERI 값을 계산하므로 버스트한 입력이 유입되었을 경우 이상적인 부하지수를 계산할 수 없다. 예를 들면 백그라운드 부하로 설정된 자기 유사한 트래픽의 입력은 50Mbps로 설정되어 있지만 H 값이 증가함에 따라 입력은 0~150Mbps로 아주 다양하게 유입되고 그 변동이 심하고, 변화의 주기도 아주 빨라진다. 따라서 입력의 변동이 심하지 않을 경우에는 정확한 입력율을 판단할 수 있지만, 입력 변동이 심하게 되면 정확한 입력율을 판단할 수 없다. 그러므로 입력율이 심하게 변할수록 부하지수의 산출이 어려워지게 되고, ERICA 스위치 알고리즘에서는 이와같은 부정확한 부하지수의 산출 때문에 정확한 ER 값



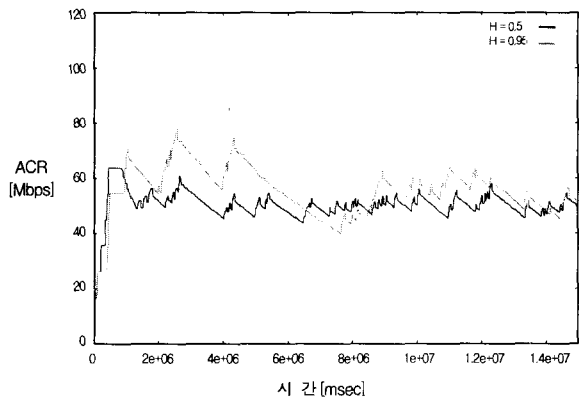
(a) EFCI ACR



(b) EPRCA ACR



(c) ERICA ACR



(d) NIST ACR

(그림 2) 스위치 알고리즘에 따른 ACR 변화

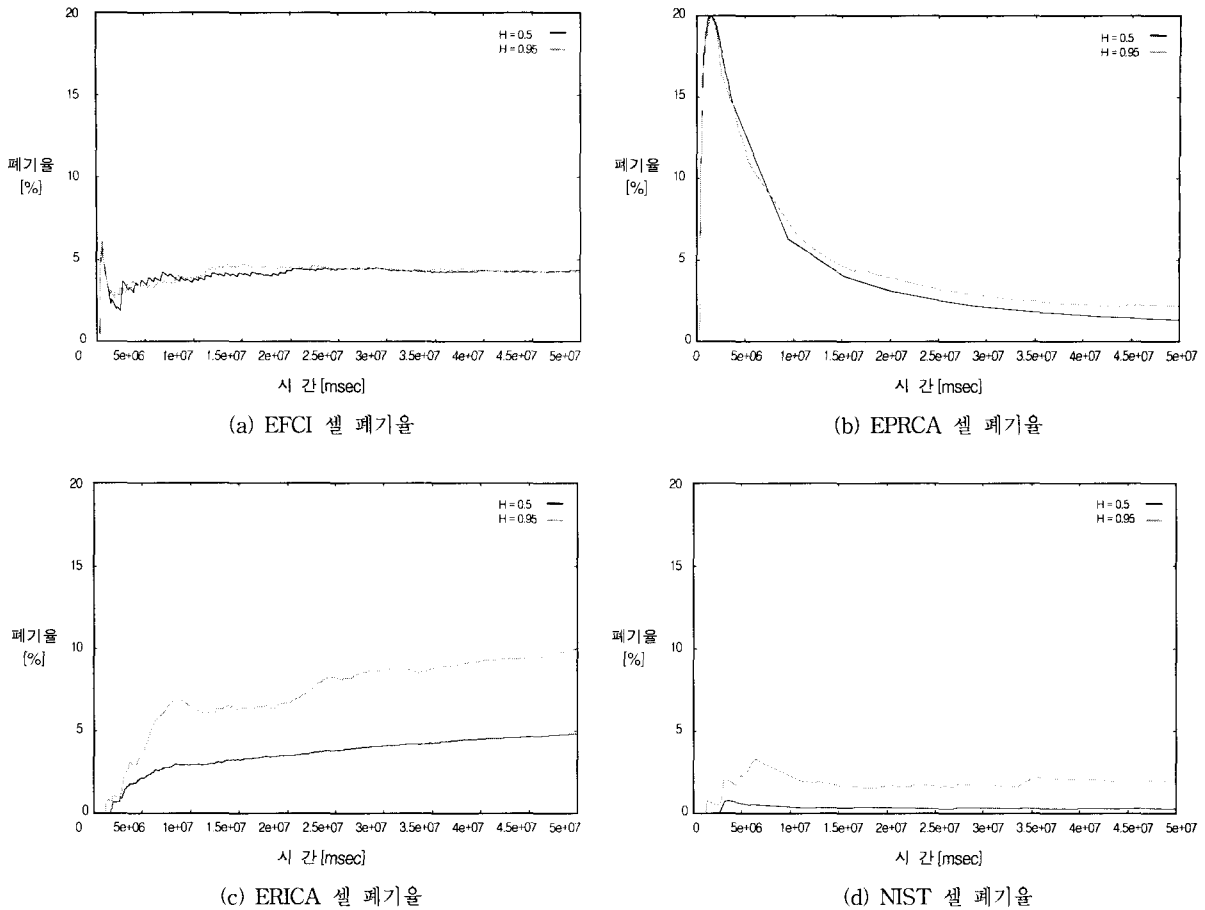
을 산출할 수 없으므로 스위치의 혼잡 상황에서도 ACR의 변동이 없게 된다. 그리고 입력율의 변동이 심할 경우 즉 $H = 0.95$ 일 때 더욱 더 부정확한 부하지수를 산출해 내므로 $H = 0.5$ 일 때 보다 더 큰 ACR 값을 나타내게 된다.

NIST 스위치 알고리즘에서는 ERICA 스위치 알고리즘과 같은 혼잡 회피 알고리즘이지만 큐 길이의 증감을 이용하여 빠르게 혼잡 상황을 파악할 수 있는 조기 혼잡 검출 방식을 사용하고 있다. 즉 스위치의 큐 길이가 임계치 Q_T 를 넘고 변화율이 양이면 혼잡 상태로 판단하고, ER 값을 설정하기 때문에 ERICA 스위치 알고리즘과 같이 혼잡의 상태를 부하지수만을 사용하여 스위치의 혼잡을 판단하는 스위치 알고리즘보다 스위치의 혼잡 상태를 더 정확하게 알 수 있으므로 버스트한 트래픽에 대해 민감하게 반응한다. (그림 2)(d)는 NIST 스위치 알고리즘을 적용한 경우의 ACR로 스위치의 큐를 이용하여 혼잡을 파악하므로 ERICA 스위치 알고리즘과 같이 동일한 혼잡 회피 알고리즘이지만 ACR의 변화가 있음을 알 수가 있다. 그리고 NIST 스위치 알고리즘은 송신원 ACR의 감소시에 스위치의 과부하 정도에 상관없이 고정된 상수인 MRF를 MACR에 곱하여 감소시키기 때문에 다른 스위치에 비해 감소시간이 크게 나타남을 알 수 있다. 또한 $H = 0.95$ 와 $H = 0.5$ 일 경우를 비교해 보면 H

$= 0.95$ 일 때가 $H = 0.5$ 일 때보다 전송율의 최대 피크치와 최소 피크치가 각각 15 Mbps씩 차이가 난다. 이것은 NIST 스위치 알고리즘의 전송율 변경시 MAIR과 MRF의 고정 상수로서 증가와 감소를 시키기 때문에 감소 시간이나 증가 시간이 커지게 되고 이것은 버스트한 트래픽에 대해서는 민감하게 반응을 하지 못하게 되는 결과로 (그림 2)(d)와 같이 ACR의 차이가 나타나게 된다.

4.3 셀 폐기율

(그림 3)은 스위치에서 각각의 알고리즘을 적용했을 경우 스위치 내 큐의 셀 폐기율을 나타낸 것이다. (그림 3)에 나타난 바와 같이 셀 폐기율은 NIST, EPRCA, EFCI, ERICA 스위치 알고리즘 순으로 우수하게 나타났다. (그림 3)(c)는 ERICA 알고리즘을 적용한 경우로 ERICA 스위치 알고리즘은 혼잡 회피 방식이므로 스위치내의 큐의 상태를 판단하는 방법이 없고, 부하지수 결정시 입력율과 목적율만을 사용하기 때문에 버스트한 트래픽에서는 현재 부하의 상태를 정확히 파악하지 못하므로 과도한 트래픽을 입력하는 결과를 낳게 되어 스위치내의 셀 폐기율이 타 방식보다 높아졌다. 그리고 $H = 0.5$ 와 $H = 0.95$ 를 비교해 보면 트래픽 버스트의 강도가 높은 경우 셀 폐기율이 더욱 더 높아짐을 알



(그림 3) 스위치 알고리즘에 따른 셀 폐기율 변화

수 있고, 스위치내의 큐에 누적된 셀들로 인해 시간이 경과할수록 셀 폐기율은 점점 더 증가함을 알 수 있다. 또한 소스들이 트래픽을 발생시킬 때 초기에는 부하지수와 공평한 공유값을 이용해 전송율을 제어하므로 타 방식에 비해 초기에 PCR 값으로 전송되는 순간적인 트래픽의 유입이 없다. 따라서 소스들이 활동한 초기에는 버퍼에 누적되는 셀들이 없으므로 셀 폐기율이 타 방식에 비해 높지 않으나 시간이 경과할수록 트래픽의 버스트에 의한 혼잡을 판단하지 못해 전송율의 변화를 가져오지 못함으로 결국 스위치내의 큐에 셀들이 과도하게 누적되어 셀 폐기율의 증가를 가져오게 됨을 알 수가 있다. EFCI, EPRCA, NIST 스위치 알고리즘은 스위치내의 부하의 상태를 스위치의 큐를 이용해 혼잡을 판단하므로 ERICA 스위치 알고리즘보다는 셀 폐기율이 낮음을 알 수 있다.

(그림 3)(a)는 EFCI 스위치 알고리즘을 적용했을 경우로 셀 폐기율은 H=0.95일 경우나 H=0.5일 경우 모두 4.13%로 일정하였다. 이것은 스위치 혼잡의 판단을 스위치내의 큐만을 이용하고, 전송율의 증가와 감소를 단순한 혼잡 상황만을 이용하므로 버스트의 강도가 큰 트래픽이나 강도가 작은 트래픽에 상관없이 셀 폐기율은 일정하게 나타난다. EFCI 스위치 알고리즘은 소스의 활동 초기에 PCR로 셀을

전송하기 때문에 초기에는 셀 폐기율이 증가하였지만 혼잡 상황을 판단한 후에는 ACR을 조절하기 때문에 셀 폐기율이 점차 감소하여 일정한 수준으로 됨을 알 수 있다.

(그림 3)(b)는 EPRCA 스위치 알고리즘을 사용했을 경우로 소스들이 활동한 초기에는 PCR로 트래픽을 스위치에 유입하여 셀 폐기율이 20%까지 높아졌음을 알 수 있다. 초기에 혼잡 상태가 된 후 소스들은 ACR의 감소를 RDF에 의해 감소시키기 때문에 초기에 스위치가 안정된 상태가 될 때까지 소스의 ACR이 서서히 감소가 되고, 이로 인해 스위치의 큐내에 과다한 셀들이 누적되어 셀 폐기율이 다른 알고리즘에 비해 높게된다. 그러나 안정 상태가 된 이후에는 공평한 공유 계산과 두 개의 큐를 이용한 혼잡 상태의 검출로 인해 과다한 셀의 유입이 적어지므로 셀 폐기율은 계속적으로 감소함을 알 수가 있다. 그리고 버스트한 트래픽의 유입이 있을 경우에도 MACR이 급격하게 감소되기 때문에 스위치 큐내에 누적되는 셀들의 시간이 길지 않으므로 셀 폐기율이 EFCI 스위치 알고리즘이나 ERICA 스위치 알고리즘보다 우수함을 알 수 있다. H=0.95와 H=0.5일 경우를 비교해 보면 H=0.95인 경우 트래픽의 버스트가 강하기 때문에 스위치내의 큐내에 누적된 셀이 지속되는 시간이 H=0.5일 경우보다 길지만 대신 트래픽 버스트의

강함으로 인한 ACR의 진동율이 낮아지기 때문에 그 만큼 스위치 큐내의 셀들이 누적되지 않는 시간이 길어지므로 $H = 0.95$ 일 경우나 $H = 0.5$ 일 경우 셀 폐기율에서는 큰 차이가 나지 않았다.

(그림 3)(d)는 NIST 스위치 알고리즘을 사용했을 경우로 NIST 스위치 알고리즘은 스위치 큐 길이의 증감을 이용하여 빠르게 혼잡 상황을 파악할 수 있는 조기 혼잡 검출 방식을 사용하기 때문에 타 방식에 비해 셀 폐기율이 월등히 우수함을 알 수가 있다. 그러나 혼잡 상황이 발생했을 때 ACR을 감소시키는 MRF를 고정된 상수로 사용하기 때문에 링크 효율면에서는 개선을 가져 올 수 있으나 트래픽 버스트가 강한 입력에 능동적으로 대처를 할 수가 없으므로 어느 정도의 셀 폐기는 생기기 되고, 이로 인해 트래픽 버스트의 강도가 큰 $H = 0.95$ 의 경우 $H = 0.5$ 일 때보다 약 1.5%의 더 큰 셀 폐기율을 나타내고 있다.

4.4 버퍼 사용률

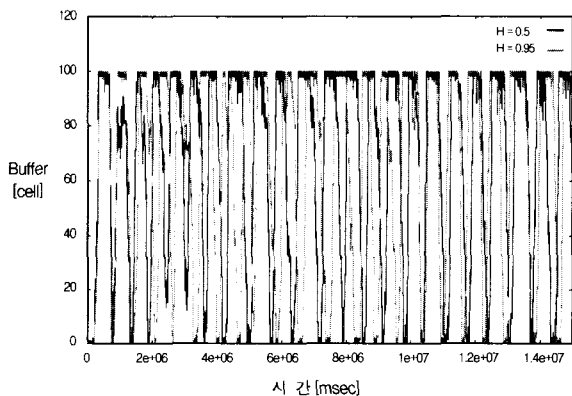
(그림 4)는 스위치에 각각의 알고리즘을 적용했을 경우 스위치 내의 버퍼 사용율을 나타내고 있다.

(그림 4)(a)는 EFCI 스위치 알고리즘을 사용했을 경우 버퍼의 사용율을 나타내고 있다. EFCI 스위치 알고리즘은

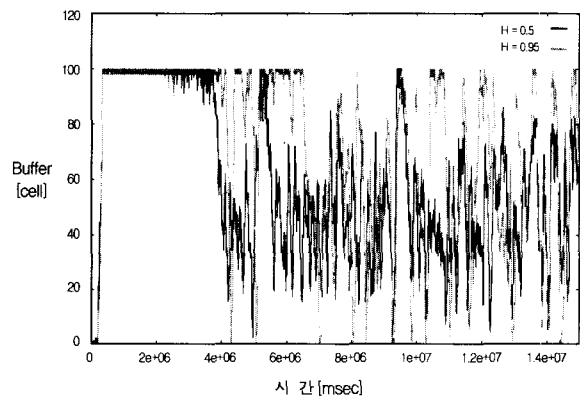
단순한 혼잡의 상황과 비 혼잡의 상황만을 파악하여 데이터를 전송하기 때문에 버퍼에 누적된 셀의 변화가 0에서 100셀까지 다양하게 나타났고, 그 변화율도 아주 큼을 알 수 있다. 또한 트래픽의 버스트 강도에 상관없이 버퍼의 사용율은 거의 동일함을 알 수 있다.

(그림 4)(b)는 EPRCA 스위치 알고리즘을 사용했을 경우 버퍼의 사용율을 나타내고 있다. EPRCA 스위치 알고리즘을 사용했을 경우에는 소스 활동 초기에는 PCR로 인한 과도한 트래픽의 입력과 ACR의 느린 감소로 인해 버퍼의 최대까지 점유가 되고 있지만 안정상태가 된 이후에는 혼잡 발생시 급격한 MACR의 감소로 인해 버퍼를 최대까지 점유하는 시간이 매우 짧아짐을 알 수 있다. 또한 $H = 0.95$ 와 $H = 0.5$ 인 경우를 비교해 보면 $H = 0.95$ 인 경우 버스트의 강도가 크기 때문에 $H = 0.5$ 일 때 보다 최대까지 버퍼를 사용하고 있는 시간이 길지만 그 만큼 ACR의 진동이 낮아지기 때문에 $H = 0.5$ 일 때 보다 버퍼 사용량의 진동이 낮음을 알 수가 있다. $H = 0.5$ 일 경우는 트래픽의 버스트 강도가 낮으므로 버퍼를 전체적으로 사용하는 시간이 극히 드물지만 ACR의 진동이 매우 빠르므로 버퍼 사용량의 진동이 $H = 0.95$ 일 때 보다 큼을 알 수가 있다.

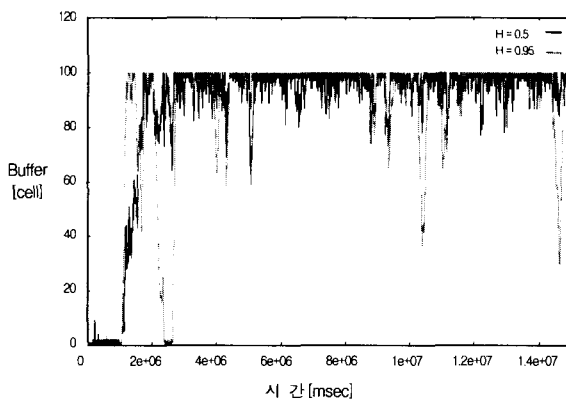
(그림 4)(d)는 NIST 스위치 알고리즘을 사용했을 경우로



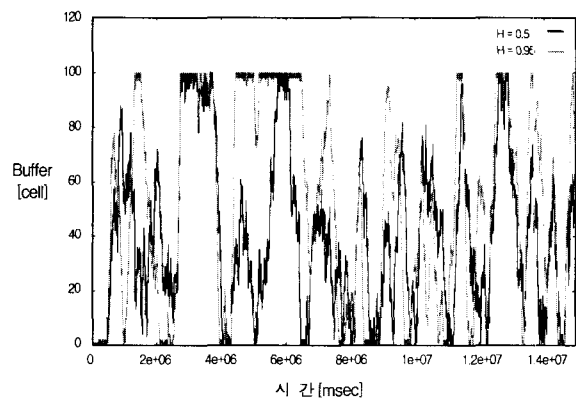
(a) EFCI 버퍼 사용률



(b) EPRCA 버퍼 사용률



(c) ERICA 버퍼 사용률



(d) NIST 버퍼 사용률

(그림 4) 스위치 알고리즘에 따른 버퍼 사용률 변화

혼잡회피 및 조기 혼잡 검출방식이기 때문에 타 알고리즘에 비해 버퍼의 최대값까지 사용하는 경우가 드물다. 그리고 ACR의 변화가 EFCI 스위치 알고리즘이나 EPRCA 스위치 알고리즘보다 작기 때문에 EPRCA 스위치 알고리즘보다 버퍼의 최대값까지 사용하는 시간이 다소 늘어나지만 대신 버퍼의 셀 변화 주기가 작아지므로 셀의 폐기가 타 알고리즘에 비해 개선되었음을 알 수가 있다. 하지만 $H=0.95$ 인 경우와 $H=0.5$ 인 경우를 비교해 보면 트래픽 버스트의 크기가 강할 경우에 셀의 폐기를 벗어날 수 없음을 알 수가 있다.

(그림 4)(c)는 ERICA 스위치 알고리즘을 사용하였을 경우로써 목표 이용율을 1보다 작은 값으로 사용하는 혼잡회피 방식을 사용하기 때문에 소스들의 활동 초기에는 버퍼의 사용량이 그다지 크지 않지만 입력 트래픽의 버스트로 인한 부하지수의 잘못된 판단으로 ACR의 변화를 주지 못해 일정 시간이 지난후 지속적으로 최대값까지 버퍼를 사용하고 있음을 알 수가 있다. 또한 $H=0.95$ 와 $H=0.5$ 인 상태를 비교해 보면 $H=0.5$ 일 경우 평균 95셀, $H=0.95$ 일 경우 평균 100셀로서 트래픽의 버스트가 강할수록 ERICA 스위치 알고리즘에서는 부하지수의 판단이 그만큼 잘못됨을 알 수가 있고, 이에 따라 버퍼에서 오버플로우(overflow)가 지속됨을 알 수가 있다.

4.5 전송 효율

<표 4>는 각 알고리즘을 사용하였을 경우 전송 효율을 나타낸 것이다. 각 알고리즘을 비교해 보면 알고리즘에 따라 트래픽 버스트의 크기가 차이가 나더라도, 또한 알고리즘의 차이가 나더라도 수신된 셀의 수는 큰 차이가 없었다. 그러나 폐기된 셀의 수를 살펴보면 알고리즘에 대해서도 많은 차이가 나지만 트래픽 버스트의 강도에 따라서도 많은 차이를 보였다. EFCI 스위치 알고리즘은 스위치의 혼잡 상황만을 고려하여 ACR을 조정하기 때문에 트래픽의 버스트 강도에 따라서 전송 효율에 큰 차이가 없었지만 다른 알고리즘은 전송 효율이 4%~20% 이상의 차이를 보이고 있다. 이를 바탕으로 분석하면, 전체적으로 보았을때 NIST 스위치 알고리즘이 타 알고리즘보다 우수한 전송 능력을 보였고 EPRCA, EFCI, ERICA 스위치 알고리즘 순으로 전송 능력이 우수하게 나타났다. 또한 버스트의 강도에 많은 영향을 받는 알고리즘은 ERICA 스위치 알고리즘으로 나타났고 NIST, EPRCA, EFCI 스위치 알고리즘 순으로 트래픽 버스트에 대한 영향을 적게 받는 것으로 나타났다. 이러한 차이점은 알고리즘 부하의 상태를 측정할 때 검출방법에 차이가 나기 때문이며, 또한 ACR을 변동시킬 때 어떠한 파라미터를 사용하는가에 따라 차이가 난다. ERICA 스위치 알고리즘이 전송효율과 트래픽 버스트의 강도에 능동적으로 대처를 하지 못하는 이유는 혼잡 상태를 검출할 때 큐의 상태를 참조하는 것이 아니고, 단순히 부하지수만을 고려하기 때문에 트래픽의 버스트에 대처를 하지 못하고 전

송효율도 타 알고리즘에 비해 많이 낮다는 것을 알 수 있다. 그러나 EFCI, EPRCA 및 NIST 스위치 알고리즘은 부하의 상태를 검출할 때 큐의 길이를 참조하기 때문에 78% 이상의 전송효율을 나타내고 있다. 또한 EPRCA 스위치 알고리즘은 스위치의 혼잡시 ACR을 급격하게 낮추기 때문에 EFCI 스위치 알고리즘보다는 전송효율이 우수함을 알 수가 있으며, 이러한 스위치 알고리즘의 특성으로 인해 버스트의 강도가 차이가 나더라도 전송효율에는 큰 차이가 없었다. NIST 스위치 알고리즘은 스위치의 혼잡 상황을 조기에 검출하기 때문에 타 방식에 비해 아주 우수한 전송효율을 나타내고 있다. 그러나 ACR 변동시 고정된 MRF 값을 사용함으로 인해 ACR의 변동이 능동적으로 입력 트래픽에 대해 대처를 못하므로 트래픽의 버스트 강도가 약한 $H=0.5$ 일 경우에는 아주 우수한 전송효율을 나타내고 있지만 버스트의 강도가 커질수록 전송효율에 대해 많은 차이가 난다는 것을 알 수가 있다.

<표 4> 알고리즘에 따른 전송효율

측정값	EFCI		EPRCA		ERICA		NIST	
	H=0.5	H=0.95	H=0.5	H=0.95	H=0.5	H=0.95	H=0.5	H=0.95
폐기된 셀의 수	15639	15124	4563	7696	17835	38661	914	6615
수신된 셀의 수	57479	57405	58783	59089	58547	55682	58433	58487
전송효율 (%)	78.611	79.198	92.789	88.476	76.650	59.021	98.459	89.834

5. 결 론

본 논문에서는 기존의 대표적인 ATM 스위치 알고리즘 중 EFCI, EPRCA, ERICA, NIST 스위치 알고리즘에 대해 자기 유사한 트래픽을 적용하여 반응을 분석하였다. 기존에 많은 연구 논문에서는 포아송이나 마코비안을 기반으로 하는 큐잉 모델을 사용하였기 때문에 실제 버스트한 트래픽을 적용했을 경우와는 많은 차이가 있다. 따라서 본 논문에서는 기존의 알고리즘에 버스트한 트래픽을 적용하여 실제 운영중인 네트워크와 유사한 모델을 설정한 후 버스트한 트래픽에 대한 반응성을 ACR, 버퍼 사용율, 셀 폐기율, 전송 효율 등으로 나누어 분석을 하였다. 분석 결과에서 나타났듯이 전송효율 면에서는 NIST 스위치 알고리즘이 가장 우수하게 나타났고 EPRCA, EFCI, ERICA 스위치 알고리즘 순으로 전송효율이 우수한 것으로 나타났다. 그리고 트래픽 버스트의 정도에 따라서는 EPRCA 스위치 알고리즘이 트래픽 버스트의 강도가 심한 경우나 약한 경우에 상관없이 일정한 성능을 나타내고 있었고, 또한 EFCI 스위치 알고리즘도 트래픽 버스트의 강도에 상관없이 일정한 성능을 나타내고 있었다. 그러나 ERICA, NIST 스위치 알고리즘은 트래픽 버스트의 강도에 따라 심한 차이점을 나타내고 있음을 알 수가 있다. 이러한 분석으로 알고리즘의 성능은 스위치의 혼잡을 판단할 때에는 단순한 부하지수만을 사용하는 알고리즘보다 스위치의 큐 상태를 판단하여 혼잡의 정도를 판

단하는 알고리즘이 버스트한 트래픽에 대한 반응이 우수하다는 것을 알 수가 있고, ACR의 변화를 줄 때 고정된 상수의 파라미터보다 동적인 파라미터의 값이 요구됨을 알 수가 있다.

본 논문에서는 단순한 모델을 사용하였기 때문에 ERICA, NIST, EPRCA 스위치 알고리즘의 장점인 공평한 공유에 대한 성능을 분석하지는 못하였다. 그러나 이러한 연구결과를 통해 알 수 있는 것은 기존의 포아송이나 마코비안 큐잉 모델을 사용하였을 경우 실제 성능에는 많은 차이가 남을 알 수가 있다.

참 고 문 헌

- [1] 김창호, 김동일 외, "실시간 운영중인 Ethernet 트래픽의 장기간 의존성 및 Self-similar 특성에 관한 연구", 한국해양정보통신학회 추계종합학술발표대회, Vol.3, No.2, pp.331-335, 1999.
- [2] 진성호, 임재홍, "실시간 운영중인 네트워크 상에서 Self-similarity 특성 및 Hurst 파라미터 측정", 한국해양정보통신학회 추계종합학술발표대회, Vol.3, No.2, pp.266-279, 1999.
- [3] William Stalling, *High-Speed Network*, Prentice Hall, pp.182-207, 1998.
- [4] Peyton Z. Peebles, JR. *Probability Random Variables and Random Signal Principles*, McGraw Hill, pp.134-198, 1993.
- [5] W. Willinger, M. S. Taqqu, R. Sherman, "Self-similarity through high-variability-statistical analysis of Ethernet of LAN traffic at the source level," *IEEE/ACM Transactions on Networking*, Vol.5, No.1, pp.71-86, 1997.
- [6] Will E. Leland, Walter Willinger, Bellcore, Murad S. Taqqu, Daniel V. Wilson "On the Self-similar Nature of Ethernet Traffic," *Computer Communication Review*, Vol. 25, No.1, pp.203-214, 1995.
- [7] Patrick Droz, Jean-Yves Le Boudec, "A High-Speed Self-Similar ATM VBR Traffic Generator," *Proceedings of the Globecom*, Vol.1, pp.586-590, 1996.
- [8] V. Paxson, S. Floyd, "Wide-area traffic : The failure of Poisson Modeling," *IEEE/ACM Transaction on Networking*, Vol.3, pp.226-244, 1994.
- [9] The ATM Forum Technical Committee, "ATM Forum Traffic Management Specification Version 4.0," ATM Forum af-tm-95-0056.000, Mar., 1996.
- [10] R. Jain, "Congestion Control and Traffic Management in ATM Network : Recent Advances and Survey," *Computer Networks and ISDN System*, Oct., 1996.
- [11] L. Roberts, "Enhanced PRCA (Proportional Rate Control Algorithm)," *ATM Forum/94-0735RI*, Aug., 1994.
- [12] N. Golmie, Y. Chang and D. Su, "NIST ER Switch Mechanism (An Example)," *ATM Forum/95-0695*, June, 1995.
- [13] R. Jain, "Congestion Control in Computer Networks : Trends and Issues," *IEEE Network*, pp.24-30, May, 1990.
- [14] Raj Jain, Shiv Kalyanaraman, Ram Viswanathan, "The OSU Scheme for Congestion Avoidance in ATM networks Using Explicit Rate Indication," *Proceedings WATM '95 First Workshop on ATM Traffic Management*, Paris, Dec., 1995.



진 성 호

e-mail : deodato@korea.com

1994년 동의대학교 전자통신공학과 (공학사)

1998년 동의대학교 대학원 전자공학과 (공학석사)

2001년 한국해양대학교 대학원 전자통신공학과 수료(박사수료)

관심분야 : 프로토콜 성능분석, 트래픽 엔지니어링, ATM



임 재 홍

e-mail : jhyim@kmaritime.ac.kr

1986년 서강대학교 전자공학과(공학사)

1988년 한양대학교 대학원 전자공학과 (공학석사)

1995년 한양대학교 대학원 전자공학과 (공학박사)

1995년~현재 한국해양대학교 전자통신공학과 부교수

관심분야 : 그룹웨어, 분산처리, 초고속통신망