

사설망인터페이스(PNNI)의 효율성 향상을 위한 ABR 서비스 제어

An ABR Service Control for Improving Efficiency of PNNI

김변곤*†

Byun-Gon Kim*†

전북대학교 전자정보공학부*

Div. of Electronics and Information Engineering, Chonbuk University

Abstract : In this paper, we analyze previous flow control algorithm for serving ABR traffic, and then propose the algorithm which calculate fair transmission rate and control efficiently ABR traffic using VS/VD switch which has an effect on PNNI environment with long delay. For getting the transmission rate, the proposed algorithm use FMMRA as ER scheme which calculate exactly the fair share. And, in large delay state, we can obtain fair cell sharing by calculating transmission rates which obtained by transmitted queue length information of VD to VS for serve and drain cell in queue

Through the computer simulation, we evaluate the performance of proposed algorithm. According to the results, the proposed algorithm shows good performance.

Key words : PNNI, VS/VD, traffic control

요약 : 본 논문에서는 ABR(Available Bit Rate) 트래픽의 흐름제어를 방식들을 분석하고, 지연이 큰 PNNI망에서 VS/VD 스위치를 이용하여 효율적으로 ABR 트래픽을 제어하여, 공평한 전송율을 계산할 수 있는 알고리즘을 제안하였다. 제안된 알고리즘에서는 전송율을 계산하기 위하여 정확하고 공평한 분배를 계산할 수 있는 ER(Explicit Rate) 방식의 FMMRA(Fast Max-Min Rate Allocation)를 사용하였다. 그리고, 지연이 많이 된 상태에서도 큐에 있는 셀을 서비스하고 드레인하기 위하여, VD에서 VS로 전달된 큐 길이 정보를 이용하여 전송율을 계산함으로써 공평한 셀 서비스 분배를 할 수 있었다. 컴퓨터 시뮬레이션을 통하여 제안된 알고리즘의 성능을 평가하였으며, 그 결과 만족할 만한 성능을 얻을 수 있었다.

주제어 : PNNI, VS/VD, 트래픽 제어

† 교신저자 : 김변곤(전북대학교 전자정보공학부)

E-mail : togonee@chonbuk.ac.kr

TEL : 063-270-7126, 016-642-0927

1. 서론

최근 인터넷을 사용하는 인구는 매우 빠른 속도로 급증하고 있으며, 사용자 층이나 서비스의 종류도 다양해지고 있다. 또한, 감성공학의 관련연구도 다양한 분야에서 광범위하게 이루어지고 있다. 이러한 연구들은 네트워크를 통하여 정보를 공유할 수 있고 활용될 수 있을 것이다. 또한, 서로 다른 지역에서 이루어지는 연구들을 네트워크를 통하여 연결함으로써 종합적인 통합 연구가 가능하다. 그리고 네트워크 기술은 사용자에게 서비스의 쾌적성 및 편의성을 제공할 수 있어야 하며, 사용자 입장에서 서비스의 안정성 및 신뢰성을 보장받을 수 있어야 한다. 또한, 실시간 서비스 등의 멀티미디어 서비스를 매우 만족스럽게 서비스받을 수 있고, 지연에 따른 시각과 청각감성을 최대한 보장받을 수 있어야 한다. 이러한 다양한 요구 조건과 특성을 갖는 사용자 중심의 서비스를 제공하고, 인터넷과 같은 버스트(burst)하고 예측 불가능한 자원의 효율적인 사용을 위해서 트래픽 흐름제어 연구가 필요하다.

ABR 클래스의 트래픽 흐름제어는 EPRCA(Enhanced Proportional Rate Control Algorithm)를 기반으로 하는 rate-based 방식을 사용하고 있다[1]. Rate-based 방식에는 폭주가 발생했을 때 이진케환(binary feedback) 정보를 이용하여 폭주를 제어하는 케환 기법과 사용 가능한 대역폭을 근사적으로 계산하여 분배하는 ER(Explicit Rate) 케환 기법이 있다.

EFCI 흐름 구조는 간단한 구조이나 망 상황에 망 상황에 신속히 대처하지 못하여 사용 가능한 대역폭을 공평하게 분배하는 능력이 떨어지는 단점이 있고, ER 흐름 구조는 대역폭을 모든 VC(Virtual Connection)들에게 공평하게 분배하고 폭주 해결 능력이 뛰어나다는 장점이 있으나, 각 VC에 액세스하여 fair share에 해당하는 최적의 전송률을 계산함으로써 계산량이 많고 이로 인하여 빠른 중앙처리장치를 필요로 하며 스위치가 복잡하며, 가격이 비싸다는 단점이 있다. 또한 사용하는 알고리즘에

따라 성능에 많은 차이가 난다.

이에 비해 VS/VD(Virtual Source Virtual Destination) 흐름 제어 구조는 EFCI 구조의 케환 루프를 여러 부분으로 분할하여 흐름 제어 루프를 줄임으로써 빠른 응답 특성을 가지고 이로 인해 폭주를 해결하는 능력이 빠르고, 또한 동적인 망 환경에서도 적용이 뛰어나다는 장점을 가지고 있다. 그러나, 스위치 내에서 VS와 VD 사이에 RM셀의 정보가 서로 유기적·협동적으로 상호 교환이 이루어져야 하고, 이러한 동작을 위해 스위치가 다소 복잡하다는 단점이 있으나, EFCI의 단점을 보완하고 ER의 장점을 수용하는 유연한 흐름 구조이다[2].

PNNI(Private Network-to-Network Interface)망은 계층적으로 구성되어 대규모 망으로의 확장이 가능하며 각 레벨마다 여러 개의 PG(Peer Group)으로 구성되어 있다. PG의 네트워크 정보는 각 그룹 내의 PGL(Peer Group Leader)에서 요약되어 PNNI망의 모든 노드에 플러딩된다. PNNI망의 계층적 구조에 적합한 ABR 트래픽 흐름제어 방식은 VS/VD 구조를 이용하는 것이다. 즉, VS/VD 구조는 네트워크를 구획하여 부하를 분산시킬 수 있는 특성을 가지고 있기 때문에 효율적으로 PG간의 트래픽 흐름제어를 수행할 수 있다.

본 논문에서는 PNNI망 구조에 적합한 VS/VD(Virtual Source Virtual Destination) 방식을 이용하여, ABR 흐름제어에 있어 사용 가능한 대역폭을 정확하게 계산하여 공평하게 분배하는 작업과 함께 큐의 길이 정보를 소스에 보내도록 함으로써 소스가 네트워크의 상황을 예측하여 과도상태 오버로드(transient overload)를 피하면서 빠르게 목표 전송률에 도달하는 알고리즘을 제안하였다.

제안한 흐름제어 구조에서는 사용 가능한 대역폭을 정확하게 계산하여 각 VC에 공평하게 배분하는 ER 방식을 적용하였다. 이때 정상상태(steady state)에서 각 VC의 큐에 쌓이는 셀을 효과적으로 배출(drain)시키기 위하여 큐의 상태를 감시하며, 수신측 스위치에서의 큐의 길이 정보를 추출하고 이를

이용하는 방안에 대해 분석하였다.

본 논문의 구성은 서론에 이어 제2장에서는 기존에 VS/VD에 적용되었던 ERICA(Explicit Rate Indication for Congestion Avoidance)에 대해 고찰하고, 제3장에서는 공평한 셀 서비스를 할 수 있는 알고리즘을 제안하였다. 제4장에서는 시뮬레이션에 사용된 환경과 결과를 통해 제안된 알고리즘의 성능을 평가하였고, 끝으로 제5장에서 결론을 내렸다.

2. VS/VD 스위치구조에 대한 기존의 ER 알고리즘

2.1 VS/VD 스위치에 ERICA 알고리즘의 적용

그림 1은 출력 버퍼형 non-VS/VD 스위치의 단일 포트에 대한 큐잉 모델을 나타낸다. 하나의 포트는 ABR VC를 위해 하나의 클래스 큐를 가지고, 출력 포트에 향하는 모든 셀들은 FIFO(First In First Out) 방식으로 클래스 큐에 들어간다.

그림 1에서 s_{ij} 는 $node_i$ 로 향하는 VC $_j$ 의 입력율이고, r_{ij} 는 클래스 큐로 진입하는 입력율이다. 노드는 입력에서 출력포트로 셀을 단순히 스위칭하기 때문에 $s_{ij} = r_{ij}$ 로 볼 수 있다.

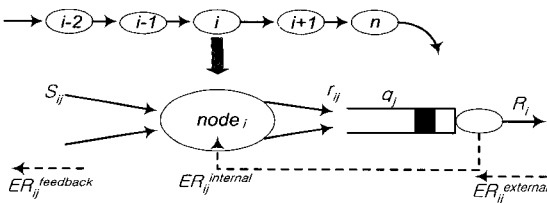


그림 1. non-VS/VD 스위치를 위한 큐잉 모델

R_i 는 ABR이 사용 가능한 전체 비트율 또는, $node_i$ 에서 클래스 큐의 출력율이고 q_i 는 클래스 큐의 큐 길이이다. N 은 링크를 공유하는 ABR VC의 수이고 F_i 는 링크의 목표 이용율이고 $0 < F_i < 1$ 사이의 값을 가진다. VC 사이에 할당된 링크율은

$\sum_{j=1}^N r_{ij} \leq F_i R_i$ 이고 클래스 큐로 진입하는 전체 입력율을 원하는 $F_i R_i$ 로 만드는 것이 스위치의 목표이다. $ER_{ij}^{internal}$ 는 노드에서 내부 혼잡을 바탕으로 계산된 ER값이며 스위치가 VC $_j$ 에게 쓸 수 있도록 허용하는 비율이다. $node_i$ 는 하향흐름 $node(i+1)$ 로부터 비율할당 정보를 받아(그림 1에서 $ER_{ij}^{external}$, 상향흐름 $node(i-1)$ 에게 퀘환 정보를 제공한다.

$$ER_{ij}^{feedback} = \text{Min}(ER_{ij}^{internal}, ER_{ij}^{external}) \quad (1)$$

$node(i-1)$ 에서 $ER_{ij}^{feedback}$ 은 $ER_{(i-1)j}^{external}$ 로 수선되고, $node(i-1)$ 은 유사한 방법으로 VC $_j$ 에 대한 비율 계산을 수행한다.

내부의 정확한 비율 계산은 로컬 스위치 상태만을 근거로 계산된다.

ERICA와 같은 방식에서는 정확한 비율 계산을 위해 몇 가지 인자를 사용하며 ERICA 알고리즘은 클래스 큐에 대한 전체 입력율, 링크의 목표 이용율, 링크를 공유하는 VC의 수 등을 다음 퀘환 주기에서 각 VC의 동작점(operation point)을 결정하기 위해 사용한다.

$$ER_{ij}^{internal} = \text{fn}(\sum_j r_{ij}, F_i R_i, N) \quad (2)$$

정상상태에서, ERICA 알고리즘은 $\sum_j r_{ij} = F_i R_i$ 를 유지한다. 그리고 과도 오버로드에 따른 큐의 축적은 $(1 - F_i)R_i$ 의 비율로 배출된다. 결과적으로 ERICA는 총 $F_i R_i$ 만큼 비율로 할당하고, 100 F_i %만큼 정상상태의 출력 링크 이용율을 얻는다.

2.2 FMMRA

다른 알고리즘보다 비교적 계산 복잡도가 적고 메모리 사용량이 적은 FMMRA 알고리즘이 VS/VD에 적합하다. FMMRA에서 사용되는 공평 할당량의 계

산식은 다음과 같이 max-min 공평 할당량으로 알려진 공식에서 유도한다[3].

$$\begin{aligned} \text{Advertised rate, } r &= \text{Fairshare of link } l \quad (3) \\ &= \frac{C - C_b}{N_l - \bar{N}_l} = \frac{C_{nb}}{N_{nb}} \end{aligned}$$

여기에서

C : 링크 l 을 통하여 ABR 서비스를 위해 사용될 수 있는 전체 가용대역폭

C_b : 링크 l 을 지나가는 병목이 일어난 VC의 전체 대역폭

C_{nb} : 병목이 일어나지 않은 VC가 사용할 수 있는 사용되지 않은 가용대역폭

N_l : 링크 l 에서 활성상태에 있는 VC의 수

\bar{N}_l : 병목이 일어난 VC의 수

N_{nb} : 병목이 일어나지 않은 VC의 수를 나타낸다.

공평 할당율이 γ_{old} 일 때 병목 상태는 $\beta_{i,old}$ 이고 대역폭은 $\lambda_{i,old}$ 인 VC에서 VC i 가 병목 상태이면 $\beta_{i,old}$ 는 1이고 병목 상태가 아니라면 0이다. 이때 VC의 상태가 새롭게 바뀌어 $\beta_{i,new}$ 가 되고 대역폭은 $\lambda_{i,new}$ 가 된다면, 새로운 공평 할당율 γ_{new} 는 다음 식과 같이 VC의 상태가 변할 때마다 명시율을 계산한다.

$$\begin{aligned} \gamma_{new} &= \frac{C_{nb,new}}{N_{nb,new}} = \frac{C_{nb,old} + \lambda_{i,old} - \lambda_{i,new}}{N_{nb,old} + \beta_{i,old} - \beta_{i,new}} \\ &= \frac{C_{nb,old} - (\lambda_{i,new} - \lambda_{i,old})}{(N_{l,old} - \bar{N}_{l,old}) - (\beta_{i,new} - \beta_{i,old})} \quad (4) \\ &= \frac{C_{nb,old} - \Delta\lambda}{N_{l,old} - (\bar{N}_{l,old} + \Delta\beta)} \end{aligned}$$

VC상태는 각 VC마다 다음 두 가지 변수 β_i 와 λ_i 에 기록된다.

$$\begin{aligned} \beta_i &= \beta_{i,new} \\ &= \begin{cases} 1, & \text{if BRM_cell_ER} < \gamma \\ 0, & \text{otherwise} \end{cases} \end{aligned}$$

$$\begin{aligned} \lambda_i &= \lambda_{i,new} \\ &= \begin{cases} \text{BRM_cell_ER}, & \text{if BRM_ER} < \gamma \\ 0, & \text{otherwise} \end{cases} \end{aligned} \quad (5)$$

$N_{l,old} - (\bar{N}_{l,old} + \Delta\beta) = 0$ 일 때, 모든 연결은 병목 연결이 되고, 이 때의 명시율은 사용하지 않는 대역폭 $C_{nb,old} - \Delta\lambda$ 로 정한다.

명시율의 초기 조건은,

$$\gamma_{new} = \begin{cases} C, & \text{if } N_{l,new} = 0, \\ \frac{C_{nb,old} - \Delta\lambda}{N_{l,new} - \bar{N}_{l,new}}, & \text{if } N_{l,new} > \bar{N}_{l,new}, \\ C_{nb,old} - \Delta\lambda, & \text{if } N_{l,new} = \bar{N}_{l,new}. \end{cases} \quad (6)$$

이고 링크 l 의 활성 연결(active connection) 수가 변하지 않았다면, $\bar{N}_{l,new} = \bar{N}_{l,old} + \Delta\beta$ 이고 $N_{l,new} = N_{l,old}$ 가 된다.

사용하지 않은 대역폭 $C_{nb,old}$ 는 이전 명시율 (γ_{old})을 사용하여 다음과 같이 구할 수 있다.

$$C_{nb,old} = \begin{cases} r_{old}(N_{l,old} - \bar{N}_{l,old}), & \text{if } N_{l,old} > \bar{N}_{l,old}, \\ r_{old}, & \text{if } N_{l,old} = \bar{N}_{l,old}. \end{cases} \quad (7)$$

ER 마킹을 한 후에 명시율을 갱신하는 업데이트 명시율, γ_U 와 ER 마킹을 위해 사용된 명시율로서 제공된 명시율(Offered Advertised Rate), γ_O 는 다음 값들을 사용하여 계산한다.

$$\gamma_U \begin{cases} \Delta\beta & = \beta_{i,new} - \beta_{i,old} \\ \Delta\lambda & = \lambda_{i,new} - \lambda_{i,old} \\ N_{l,new} & = N_{l,old} \\ \bar{N}_{l,new} & = \bar{N}_{l,old} + \Delta\beta \end{cases} \quad (8)$$

$$\gamma_O \begin{cases} \Delta\beta & = 0 - \beta_{i,old} \\ \Delta\lambda & = 0 - \lambda_{i,old} \\ N_{l,new} & = N_{l,old} \\ \bar{N}_{l,new} & = \bar{N}_{l,old} - \beta_{i,old} \end{cases} \quad (9)$$

3. VS/VD 스위치 구조에 대한 제안된 ER 알고리즘

ERICA 알고리즘은 모든 커넥션과 버퍼 요구에 대해 원하는 공평성을 얻기 위한 근본적인 제한을 가지는데, 늦게 시작된 커넥션은 균등하게 링크를 공유하나 최대-최소 공평 전송률을 얻지 못한다. 그리고 만약 원하는 목표 전송률이 최대 링크 전송률에 근접하게 설정되면, 전이기간 동안 큐가 급격히 증가하고 그 결과 막대한 셀 손실을 가져온다.

이러한 제한을 보완하는 ERICA+ 알고리즘은 활성화된 VC 개수의 산정과정에서 누락을 방지하기 위해 지수적으로 감소하는 가중평균을 사용하여 $active_k[v] = \alpha_{decay} \times active_{k-1}[v]$ 로 설정한 값을 사용하나 이 또한 ERICA 알고리즘 성능의 개선 정도에 비해 구현이 복잡하다는 단점이 있다.

FMMRA는 위에서 설명된 ERICA처럼 때때로 Max-Min Fair Share를 달성하지 못하는 문제점도 없고 빠르게 목표 전송률에 도달하나 아직까지는 VS/VD에 구현된 사례가 없었다.

특히, ABR 트래픽 흐름 제어에서 무엇보다 중요한 것은 트래픽의 수율(throughput)과 빠른 응답 특성, 이용 가능한 대역폭 변화나 VC 수의 변화 등 동적인 PNNI망 환경에 신속하게 대처하는 적응성, 버퍼의 안정성, 낮은 셀 손실을 들 수 있다. 이러한 요구 조건을 만족하는 흐름 구조로 VS/VD 구조가 있다.

또한 ER 방식의 제어는 빠른 폭주 해결 능력과 공평한 대역폭사용과 망자원의 효율적인 이용이라는 측면에서 장점이 있지만 계산량이 많고 스위치가 복잡해지기 때문에 고가라는 단점이 있다. 따라서 스위치 설계자는 스위치 구조를 단순화시키고 계산량을 줄이는 알고리즘을 구현할 필요가 있다.

따라서, 본 논문에서는 이러한 필요에 따라 O(1)의 계산 복잡도로 max-min fair rate를 계산하는 FMMRA를 선택하였다. 그러나 FMMRA에서는 단

방향 정보로 ER 마킹을 하기 때문에 FRM일 경우, 목적지에 가까운 스위치일수록 더 정확한 병목(bottleneck) 정보를 얻게 된다.

BRM일 경우는 소스에 가까운 스위치일수록 더 정확한 병목 정보를 얻게 되어 병목이 일어나지 않은 VC들에게 사용되지 않은 대역을 정확하게 할당하지 못하고, 병목이 일어난 VC와 일어나지 않은 VC에게 동일한 명시율을 제공하기 때문에 출력 링크에서 잠정적인 과잉이용(over-utilization) 문제를 가지게 된다.

본 논문에서는 이 문제를 해결하기 위하여 다음과 같이 VD의 큐 길이 정보를 사용하였다.

큐의 길이 정보를 이용하는 식은 다음과 같다.

$$ACR = ER * F(\cdot) \tag{10}$$

각 VC의 큐마다 세 개의 임계치 ($0, at_1, at_2$)를 두고 이 임계치(threshold)를 넘을 때마다 RM 셀의 남아 있는 필드에 VC의 큐 길이 정보를 기입하여 소스측으로 전송하면, 소스는 이 값에 해당하는 $F(\cdot)$ 를 ER값에 곱하여 VC별로 서비스하게 될 ACR 값을 결정한다.

따라서 그림 2에서와 같이 소스는 보내고자 하는 진행 방향의 네트워크 상황을 RM 셀을 통하여 통

$$F(\cdot) = \begin{cases} m_1, & \text{if } at_1 > RM_cell_QLI(q_{ij}) > 0, \\ m_2, & \text{if } at_2 > RM_cell_QLI(q_{ij}) > at_1, \\ m_3, & \text{if } RM_cell_QLI(q_{ij}) > at_2 \end{cases} \tag{11}$$

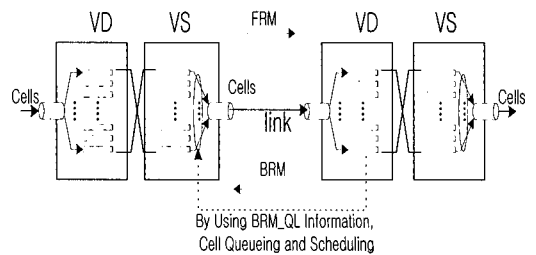


그림 2. 제안된 셀 큐잉 및 스케줄링 기법

보 받기 때문에 다른 부가적인 셀을 보내지 않고서도 큐의 크기를 줄일 수 있게 되고 또한 과도상태의 과잉이용을 극복할 수 있게 된다.

중간 노드에서 VS/VD 기능을 수행하는 스위치의 제안된 알고리즘을 적용한 기본 동작을 그림 3에 나타냈다.

그림 3의 알고리즘 흐름도에서와 같이 데이터 셀일 경우는 ACR 값으로 셀을 전송하게 되고 RM 셀일 경우는 FRM 셀을 받으면 스위치에서 계산한 fair share 값과 FRM 셀의 ER 값을 비교하여 작은 값으로 RM 셀의 ER 값을 갱신하고 VD의 큐 길이 정보를 RM 셀에 실어서 전단의 노드로 쉐환시킨다.

만약 BRM 셀을 받게되면 식 11에 따라 병목상태와 연결 정보를 수정하고 스위치에서 계산한 fair share 값과 BRM 셀의 ER 값을 비교하여 작은 값으로 RM 셀의 ER 값을 갱신하고 VD로부터 보내온 큐 길이 정보에 의해 ACR 값을 식 3.2에 따라 변경하고 갱신된 RM 셀을 다음 단의 노드로 내보냄으로써 한 주기의 셀 전송을 끝마치고 다음 셀 주기 동안 다시 그림 3의 흐름도에 따라 셀을 전송하게 된다.

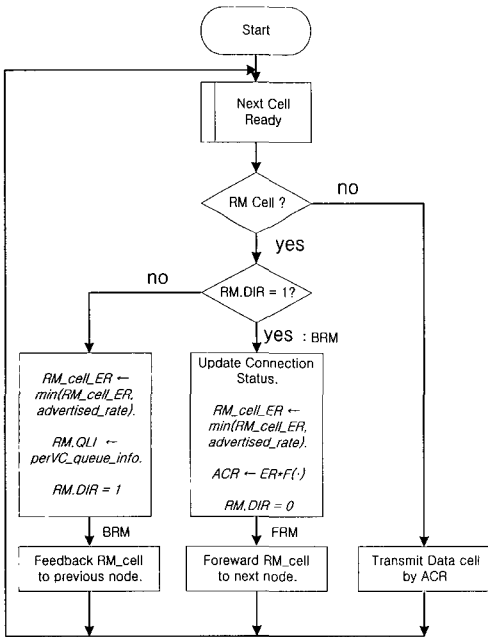


그림 3. 제안된 알고리즘의 흐름도

4. 시뮬레이션 및 성능평가

4.1 시뮬레이션 환경

본 논문에서 제안된 알고리즘의 시뮬레이션을 위한 네트워크 모델은 그림 4에서와 같이 4개의 ABR 트래픽과 1개의 백그라운드 트래픽으로 이루어져있다. 백그라운드 트래픽은 PG 1과 PG 2 사이의 PNNI 환경의 트래픽을 가정하여, 포아손 분포를 갖는 On-off 트래픽으로 평균 On 주기는 20msec, Off 주기는 80msec로 발생시켰다. On 상태일 때 전송속도는 50-80Mbps로 균등분포로 발생하고, Off때 데이터는 발생하지 않는다. ABR 트래픽 소스 파라미터는 표 1과 같다.

표 1. ABR 트래픽 파라미터

VC #	PCR	MCR	ICR	NRM
1	155 Mbps	10 Mbps	40 Mbps	32
2	155 Mbps	10 Mbps	60 Mbps	32
3	155 Mbps	10 Mbps	80 Mbps	32
4	155 Mbps	15 Mbps	100 Mbps	32

링크의 전송 속도는 모두 155Mbps이고 스위치간의 물리적인 거리는 1000km, 소스(목적지)와 스위치간의 거리는 100km로 하였다. 링크의 물리적인 지연은 km당 5 μ s로 가정하였고 스위치 내부의 동작 시간은 무시되었다. 스위치는 각각의 VC에 대하여 개별적인 큐를 가지고 있고 무한 버퍼로 가정하였다.

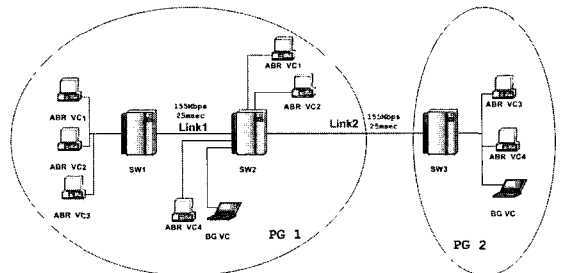


그림 4. 모의실험 네트워크 모델

4.2 성능평가 및 결과분석

그림 4, 5는 ABR VC 송신원의 ABR 트래픽 분포이다. ABR VC3는 링크 1에서 약 52Mbps로 bottleneck 되어 있고, VC4는 백그라운드 트래픽에 영향을 크게 받는다. 백그라운드 트래픽이 Off 상태일 때, VC4는 약 100Mbps로 전송할 수 있고, On 상태일 때, VC3, 4는 43Mbps로 공평하게 할당받는다.

그림 7, 8는 각각 FMMRA와 제안된 알고리즘의 SW2에서 per-VC큐의 셀 수를 보여준다. FMMRA의 경우, 백그라운드 트래픽의 On-off 상태전환이 발생할 때마다, 큐에 셀이 계속해서 쌓이는데, on 될 때 SW2에서 VC4의 버퍼가 140셀만큼 증가하는데 VC4와 SW2 사이의 링크 지연이 1msec이어서, VC4가 100Mbps에서 42Mbps로 변화되는데 변화

량이 1msec 동안 반영이 안되어서 1msec 동안 58Mbps의 잉여속도로 140셀이 쌓인다. 그 이유는 On-off 상태전환이 있을 때 앞단에서 reate 조정값이 들어간 BRM셀을 보내서 뒷단에 도착하기전에 새로운 셀을 보내기 때문에 줄어들지 못하고 계속해서 증가한다. 제안된 알고리즘의 경우, 백그라운드 트래픽이 on될 때 변화만 보이고, off일 때는 일정한 모습을 보여준다.

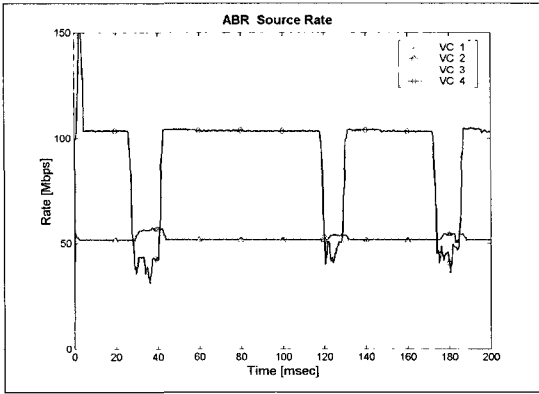


그림 5. ABR 소스 트래픽율(FMMRA)

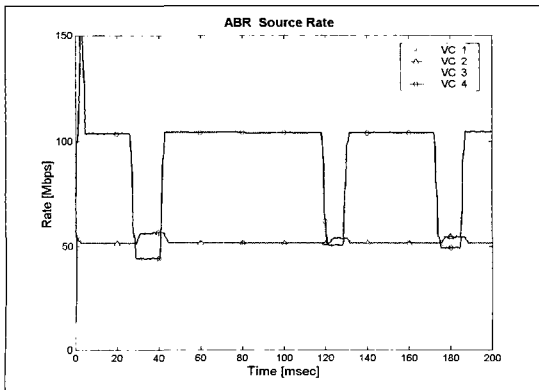


그림 6. ABR 소스 트래픽율(제안된 알고리즘)

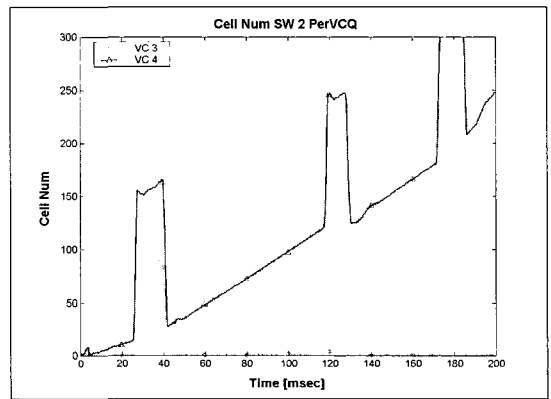


그림 7. SW2에서 per-VC큐의 셀 수(FMMRA)

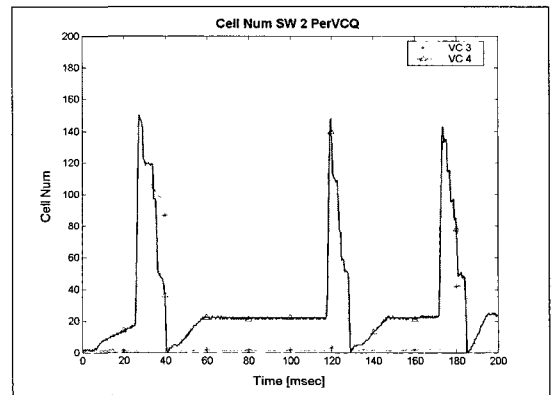


그림 8. SW2에서 per-VC큐의 셀 수(제안된 알고리즘)

그림 9, 10은 SW2에서 클래스 큐의 셀 수를 보여준다. FMMRA의 경우 셀이 완전히 줄어들지 않고 계속 남아 있고, 제안된 방식은 On 상태일 때를 제외하고는 셀이 쌓이지 않고 바로 바로 전송하는 것을 보여준다.

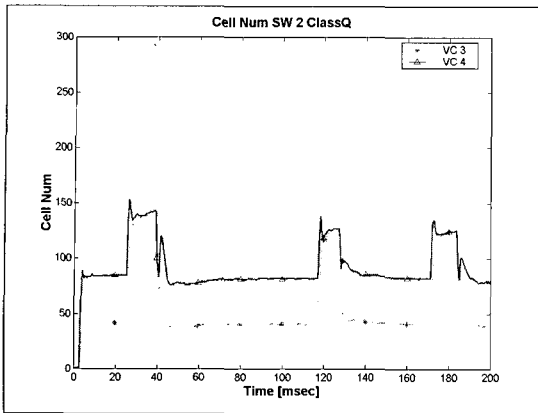


그림 9. SW2에서 class큐의 셀 수(FMMRA)

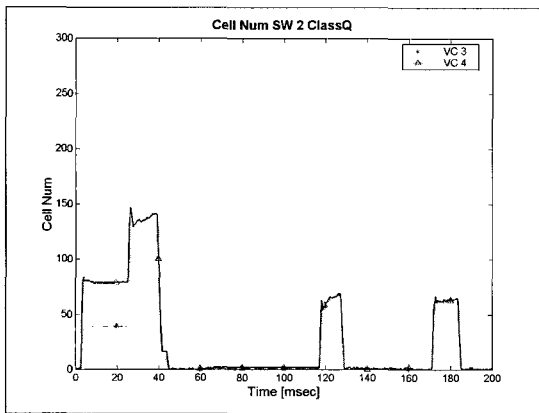


그림 10. SW2에서 class큐의 셀 수(제안된 알고리즘)

5. 결론

정보통신 기술은 사용자에게 신뢰성을 제공할 수 있어야 하며, 사용자 중심의 서비스를 제공하고, 감성네트워크 관련 연구에 활용되기 위해서는 인터넷과 같은 버스트하고 예측 불가능한 자원을 효율적으로 사용해야 한다.

본 논문에서는 PNNI와 같은 초고속 통신망에서 네트워크 자원의 효율성 향상을 위한 트래픽 제어 알고리즘을 제안하였다.

제안된 알고리즘은 기존의 순수한 FMMRA를 VS/VD 구조에 적용하였을 때 큐의 길이가 최초 RM 셀이 돌아오기까지의 RTT만큼에 해당하는 크기로 수렴하고 그 값이 줄어들지 않아 폭주 상황에

대처할 수 없고 많은 지연이 발생하는 문제점을 보완하기 위해 인접한 VD의 큐 길이 정보를 추출하고 RM 셀을 이용하여 VS에 전달하며 VS에서는 공평전송률과 큐의 길이를 같이 고려하여 전송률을 결정하는 방법을 제안하였다.

시뮬레이션 결과에서 알 수 있듯이 백그라운드 트래픽의 상태가 Off일 때, VC4는 약 100Mbps로 전송할 수 있고, On 상태일 때, VC3, 4는 43Mbps로 공평하게 할당함으로써 공평성을 만족하였고 매우 낮은 버퍼 점유율을 보여주고 있다. FMMRA의 경우, 백그라운드 트래픽의 On-off 상태전환이 발생할 때마다, 큐에 셀이 계속해서 쌓이는데, 제안된 방식은 On 상태일 때를 제외하고는 셀이 쌓이지 않는 것을 시뮬레이션을 통하여 입증하였다. 그리고 ERICA 알고리즘에서는 달성하지 못하는 Max-Min fair share를 FMMRA는 빠르게 달성하므로 VS/VD에 더 적합한 알고리즘이라고 볼 수 있다. 또한 QLI를 사용하지 않는 기존의 순수한 FMMRA 보다는 QLI를 사용한 제안된 알고리즘의 경우 버퍼 요구량에 있어서 큐를 적정수준의 작은 값을 취하기 때문에 RTT에 따른 최대 큐 성장 값만큼만 유지해도 되므로 큐를 드레인 시키지 않는 기존 알고리즘에 비해 1/2 정도의 버퍼크기만 확보해도 충분하기 때문에 버퍼 낭비를 막을 수 있고 링크의 이용을 효율적으로 사용함을 알 수 있다.

참고문헌

- [1] Sathaye, S. (1996). ATM Forum Traffic Management Specification, Version 4.0, ATM Forum Technical Committee, Mar.
- [2] Chao, H. C., & Chen, Y. C., & Chen, J. L. (2001). Interoperability of EFCI and ERICA ATM switches for ABR services with VS/VD enhancements, Computers & Electrical Engineering. 27(5), 355-366.
- [3] Liping An, & Nirwan Ansari, & Ambalavanar

- Arulamba (1998). TCP/IP traffic over ATM networks with FMMRA ABR flow and congestion control, *Computer Networks and ISDN Systems*, 29(17-18), 2091-2102.
- [4] Duan, Y. B., & Wen, C. Y., & Soong, B. H. (2002). Design of a new globally stable explicit rate controller for ABR service with saturation, *Computer Communications*, 25(8), 772-801.
- [5] Kakemizu & Mitsuaki (1999). Global PNNI Internetworking Architecture for ATM Public Networks. *Proceedings of the 1999 IEEE International Conference on Communications*.
- [6] Rogozini, A. R., & Al, E. (1999). Analysis of the Performance of a Hierarchical PNNI Network. *Proceeding of the 1999 2nd International conference on ATM*.