

# ATM 망에서 조기 공정률 할당 알고리즘의 성능 개선

정희원 김 대 일\*, 박 기 태\*, 김 남 일\*\*

## Performance Improvement of the Early Fair Rate Allocation Algorithm in ATM Networks

Dae-Il Kim\*, Ki-Tae Kim\*, Nam-Il Kim\*\* *Regular Members*

### 요 약

본 논문에서는 ABR 서비스를 지원하는 ATM 스위치를 위하여 새로운 향상된 조기 공정률 할당 알고리즘(EFRA+)을 제안한다. ABR 서비스를 위한 명시적 전송률 제어 알고리즘의 주요 논점은 모든 연결에게 최대-최소 공정률을 계산하는 것이다. EFRA+는 EFRA의 주요 특징을 계승하고, 스위치에서 잠재적 혼잡을 방지하기 위하여 연결 설정 동안에 연결 제어 정보를 이용하고, 최대-최소 공정률의 계산 방법을 향상한다. 그리하여 EFRA+는 EFRA보다 더 정확하게  $O(1)$ 의 계산 복잡도를 가지고 최대-최소 공정률을 계산한다. 이러한 공정률의 정확한 계산은 최대-최소 공정률의 수렴을 촉진하고, 우수한 과도 응답을 제공하고, 과도 상태에서 EFRA의 성능 열화를 개선한다. 정상 상태에서는 EFRA+는 전송률의 변동없이 동작하고, 다양한 ATM 망에서도 매우 안정적이다. 버퍼에서 ABR 큐 길이는 처리율의 감소가 없으면서도 크게 감소되었다. 또한 EFRA+는 혼잡 지점에서 높은 링크 이용률을 유지한다. 컴퓨터 시뮬레이션을 이용하여 EFRA+와 EFRA, ERICA+를 비교했으며, EFRA+ 알고리즘의 효율성을 알아보았다.

### ABSTRACT

In this paper, a new enhanced early fair rate allocation(EFRA+) algorithm is proposed for ATM switches supporting ABR service. The central issue of explicit rate control algorithms for ABR service is the computation of max-min fair rates for every connection. The EFRA+ inherits the main feature of the EFRA, uses the connection control information during the connection setup to prevent potential congestion in switches, and enhances the computation method of max-min fair rate. Thus, the EFRA+ computes the max-min fair rates with  $O(1)$  computational complexity more accurate than EFRA. This exact calculation of fair rates expedites quick convergence to max-min fair rates, offers excellent transient response, and improves the performance decay of the EFRA at the transient state. At the steady state, the EFRA+ operates without causing any oscillations in rates and is very robust in various ATM networks. The ABR queue length at the buffer can further be reduced without any degradation in throughput. And also the EFRA+ keeps high link utilization at the congestion point. By using computer simulation, it is compared EFRA+ with existing algorithms such as EFRA, ERICA+, and is demonstrated the effectiveness of the EFRA+ algorithm.

### I. 서 론

ATM 포럼에서는 ABR 서비스를 위하여 통신망

의 전송 특성 변화에 따른 송신단의 전송률 제어를 위하여 여러 형태의 피드백을 기술하고 있다. 이러한 피드백 정보에는 망의 혼잡 상태, 전송률 등의 내용이 포함되고, 이를 위하여 ATM 스위치는 ABR

\* 전국대학교 전자공학과

\*\* 가천길대학 전자통신과

논문번호: T00032-0824, 접수일자: 2000년 8월 24일

서비스를 위한 흐름과 혼잡 제어를 수행해야 한다. 이러한 전송률 기반 혼잡 제어 알고리즘은 최근에 많은 단체에서 제안되었다. 이들 알고리즘들의 성능은 천차만별이고, 그 수행 기능의 요구 인자들도 매우 다양하다. 이들 대부분의 알고리즘은 정확한 공정률 할당, 낮은 수행 복잡도 등을 성능의 주요한 평가 척도로서 이용하고 있다. 그러나 ABR 서비스는 트래픽 전송 시간의 대부분이 과도 상태에 놓여 있는 특성이 있다. 즉, 우선 순위가 높은 VBR 서비스는 균집성이 강해서 ABR 서비스의 가용 대역폭에 급격한 변동을 주게 되고, ABR 서비스들은 가용 대역폭을 서로 경쟁적으로 공유하므로 알고리즘은 항상 과도 상태에서 동작하게 된다. 특히 스위치를 경유하는 경로 상에서 새로운 VC(Virtual Channel)가 연결을 설정하면 통신망 자원의 예약에 따른 자원 이용의 큰 변동은 알고리즘의 성능 저하와 큰 성능 편차를 가져온다.

본 논문에서는 향상된 조기 공정률 할당(EFRA+) 알고리즘을 제안한다. EFRA+는 EFRA의 주요 장점의 계승과 동작의 단점을 보완하고, 연결 제어 정보의 조기 참조의 향상, 보다 정확한 최대/최소 공정률의 할당, 과도에서 정상 상태로의 신속한 천이, 그리고 초기 세대의 스위치와 상호 동작 등의 부문에서 성능 개선을 목표로 하고 있다. 본 논문은 II절은 ABR 전송률 기반 흐름 제어, III절은 EFRA 알고리즘을 설명하고, IV절과 V절은 새로운 향상된 조기 공정률 할당 알고리즘을 제안하고, 실험 결과를 기존 알고리즘과 비교하며, 마지막으로 VI절은 논문의 결론에 관하여 기술한다.

## II. ABR 종단간의 흐름 제어

ABR 흐름 제어는 전송률 기반, 페 루프, 그리고 각 VC별로 제어를 수행한다. 그림 1은 일반적인 종단간의 ABR 흐름 제어의 방법이다. 송신단은 현재 데이터율에 비례하여 RM (Resource Management) 셀을 순방향으로 망 내로 송신하고, 수신단은 이 셀을 수신한 후에 송신단으로 재전송한다. RM 셀은 순방향 또는 역방향으로 전송되는 동안에 ATM 스위치에서 참조 및 수정되고, 또한 혼잡 정도와 전송률 등의 망 상태 정보를 담아 전송한다. 그 결과로서 송신단은 이러한 피드백 정보에 따라 데이터의 전송 속도를 조절한다. 이러한 일련의 종단간 흐름 제어는 표준으로 정의된 송수신단의 동작을 종단 시스템이 반드시 수행해야 가능하다. 또한 망의 스

위치는 경로상의 모든 연결들에게 공정한 전송률을 할당해야 한다. 이러한 전송률 할당 방법은 수행할 기능을 명확히 기술하기 때문에, 스위치 설계와 기능의 중요한 부분이다.

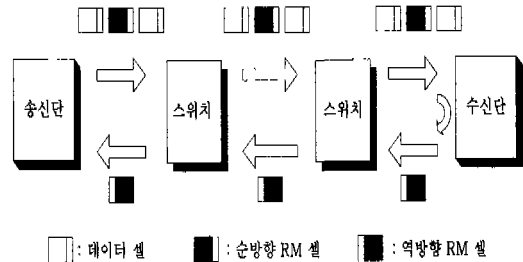


그림 1. 종단간 전송률 기반의 흐름 제어

### 2.1 스위치의 동작

ATM 스위치는 버퍼 관리 관점에서 다음과 같은 혼잡 제어 방법들 즉, 명시적 순방향 혼잡 지시(Explicit forward congestion indication), 상대적 전송률(Relative rate), 명시적 전송률(Explicit rate) 기술 중에서 최소한 한가지를 수행해야 한다. 첫째는 데이터 셀 헤더내의 EFCI 비트의 마크이며, 둘째는 순방향 RM(FRM) 또는 역방향 RM 셀(BRM) 내의 CI (Congestion indication), NI(No increase) 비트의 마크이다. 이들 방법은 복잡도는 낮지만, 불공평성, 느린 혼잡 응답, 혼잡 오차 등의 특성이 있다. 이러한 방법을 수행하는 스위치는 제 1세대로서 “Binary switch” 라고 한다. 셋째는 대역폭의 공평한 할당을 위한 복잡한 계산 메커니즘이 필요하며, 순(역)방향 RM 셀 내의 ER(Explicit rate) 필드에 계산한 공정률을 삽입한다. 혼잡 제어는 더 복잡하지만, 제 1세대보다 성능은 우수하며, “ER switch” 라고 한다. ATM 포럼에서는 스위치에 관한 동작은 다른 종류의 스위치와의 상호 연동에 필요한 기초적인 내용만 기술하고 있으며, 상세한 동작은 제조업체에 전달시키고 있다. 이러한 유연성으로 인하여 다양한 성능을 가진 스위치를 제작할 수 있다.

### 2.2 송수신단의 동작

송수신단은 ATM 스위치의 여러 혼잡 제어 방법들과 연동하는 기능을 수행해야 하며, 데이터 셀, RM 셀을 전송할 때는 설정된 ABR 파라미터 값에 따라 동작해야 한다.

- 송신단은  $ACR = ICR$  값으로 초기에 데이터를 전송하고, 망에서 채환된 BRM 셀의 정보를 참조하

여  $MCR \leq ACR \leq PCR$  범위에서 전송한다.

- 송신단은 BRM 셀의 NI와 CI 비트를 참조하여 현재 전송 ACR 값을 증가, 감소한다. 각 비트 값에 따른 ACR 계산 규칙은 표 1과 같다.

- 송신단은 최초 데이터 전송이후 FRM을 최소한 Mrm, Trm 값, 또는 (Nrm-1) 개의 데이터 셀마다 생성하여 전송한다. FRM(CCR, MCR, ER) 셀 내에는  $CCR=ACR$ ,  $MCR=MCR$ ,  $ER=PCR$  값이 할당된다.

- 수신단은 수신한 데이터 셀, FRM 셀을 조사하고, BRM 셀을 생성하여 혼잡 정보와 CI, NI 비트를 수정하여 역방향으로 송신단에 전송한다.

표 1. ACR 값의 계산 규칙

NI	CI	ACR 값
0	0	$\max[MCR, \min[ER, PCR, ACR + RIF*PCR]]$
0	1	$\max[MCR, \min[ER, ACR(1 - RDF)]]$
1	0	$\max[MCR, \min[ER, ACR]]$
1	1	$\max[MCR, \min[ER, ACR(1 - RDF)]]$

### III. EFRA 알고리즘

조기 공정률 할당, 즉 EFRA(Early Fair Rate Allocation) 알고리즘은 2000년 6월에 김대일, 김남일 등에 의하여 제안되었다<sup>[3][4]</sup>. EFRA는 신호 제어 과정에서 발생하는 연결 설정과 해제에 따른 트래픽 파라미터 정보를 각 송신단의 공정률을 계산하는 과정에 조기에 반영한다. 그 결과로서 기존의 명시적 알고리즘들에서 발생하는 긴 과도 상태의 성능 저하를 정상 상태로 빠르게 회복시킨 점이 중요한 특징이다. 또한 EFRA는 ERICA+의 특징인 정확한 공정률 할당, O(1)의 낮은 계산 복잡도, 100%의 출력 링크 이용률 등의 장점을 계승하고, ERICA+의 단점을 보완하여 ERICA+보다 스위치의 큐 길이와 전송 지연을 크게 낮추어서 전송 처리율을 높이고, 버퍼에서의 셀 손실률은 더 감소시켰다. 다음은 EFRA 알고리즘의 주요 동작 과정이다.

#### 3.1 공정 할당값(FairShare)의 계산

스위치는 주기적인 AI(Averaging Inter val) 시간 간격마다 가용 ABR 대역폭, 부하 인자(z), ABR 유입률( $\lambda_{abr}$ ), 그리고 공정 할당값을 계산한다. EFRA는 식 (1)과 같이 가용 ABR 대역폭을 유휴 대역폭인 총 ABR 대역폭과 스위치 버퍼의 ABR 큐 길이

를 감안한 큐 제어 함수  $f(Tq)$ 의 값을 승산하고, 여기에 연결 설정 과정에서 참조하여 저장해 둔 연결 제어 대역폭을 차감하여 계산한다. 연결 제어 정보 처리 과정은 3.3절에서 설명한다. 그리고 모든 VC가 공평하게 분배받는 공정 할당값을 식 (2)와 같이 계산한다.

$$\begin{aligned} \text{총 ABR 대역폭} &= \text{링크대역폭} - \text{CBR\_VBR 대역폭} \\ \text{가용 ABR 대역폭} &= \text{총 ABR 대역폭} \times f(Tq) \\ &\quad - \text{연결 제어 대역폭} \end{aligned} \quad (1)$$

$$\text{"FairShare} = \frac{\text{가용 ABR 대역폭}}{\text{총}} \text{ ABR 연결의 수} \quad (2)$$

ABR 유입률( $\lambda_{abr}$ )은 매 AI 시간 간격마다 스위치로 유입되는 ABR 트래픽량이다. 이 값은 스위치로 유입되는 ABR 트래픽의 총량으로서 매 AI 시간마다 측정된다. 부하 인자(z)는 가용 ABR 대역폭의 이용률, 즉 스위치로 유입되는 ABR 트래픽의 부하 정도를 나타낸다. 식 (3)에 의하여 계산하고, 제안자에 의하면 파라미터 상수  $\alpha$  는 0.8이다.

$$z = (1 - \alpha) \times z + \alpha \times \frac{\text{ABR 유입률}}{\text{가용}} \text{ ABR 대역폭} \quad (3)$$

#### 3.2 공정률(Fair Rate)의 할당

각 VC 별로 역방향으로 BRM 셀이 도착하면 3.1절에서 계산한 공정 할당값을 VC 별로 재조정하여 VCShare 값을 계산하고, 새로운 ER 값을 계산하여 BRM 셀 내의 ER 값과 비교한 후에 최종적인 ER 값을 역방향으로 근접한 스위치로 전송한다. 식 (4)에서 각 VC의 VCShare 값은 각 VC의 현재 전송률과 부하 인자를 계산하여 계산한다. 부하 인자가 1보다 크면 스위치는 처리 용량이 과부하 상태이므로 현재 전송률은 감소해야만 한다. CCR (Current Cell Rate)은 각 VC의 현재 전송률이다.

$$\text{"VCShare} = \frac{\text{"CCR}}{z} \quad (4)$$

최종적으로, 스위치는 식 (5)와 같이 새로운 ER 값을 계산하고, 식 (6)과 BRM 셀 내의 ER 필드에 새로운 ER 값을 대체하게 된다. 제안자에 의하면 파라미터 상수  $\delta$  는 0.1 정도의 작은 값이고, MaxAllocprev 값은 이전의 AI 시간에 계산한 최대 ER 값이다.

$$\text{If } (z > 1 + \delta) \quad ER_i = \max(\text{FairShare}, \text{VCShare})$$

$$\text{Else } ER_i = \max(\text{FairShare}, \text{VCShare}, \text{MaxAllocprev}) \quad (5)$$

$$ER_{BRM} = \min(\text{가용ABR대역폭}, ER_{BRM}, ER_i) \quad (6)$$

### 3.3 연결 제어 정보의 참조

스위치에서 역방향으로 새로운 VC가 연결 설정을 완료하면 VC의 정보 테이블은 생성된다. 이때 EFRA는 VC 별로 서비스의 트래픽 파라미터인 대역폭(B)을 식 (7)과 같이 저장하고, 주기적인 AI 시간마다 식 (8)과 같이 ABR 연결의 수는 하나를 증가하고, 새로운 서비스가 CBR, VBR이면 식 (1)의 가용 ABR 대역폭은 식 (9)와 감소된다. 반면 ABR 서비스이면 ABR 트래픽 유입률은 식 (10)와 같이 향후 예상되는 ABR 트래픽의 유입을 조기에 반영한다. 식 (9), (10)에서 연결 제어 정보의 참조 과정은 실제 송신단이 데이터를 전송하고, 각 VC의 첫 번째 데이터가 스위치에 도착할 때까지만 반영되고, 식 (7)의 연결 제어 대역폭은 즉시 삭제된다.

$$B_{cbr}[VC] = PCR, \quad B_{vbr}[VC] = SCR,$$

$$B_{abr}[VC] = \max(ICR, MCR) \quad (7)$$

$$C_{abr,new} = C_{abr,old} + 1 \quad (8)$$

$$\text{가용 } B_{abr,new} = \text{가용 } B_{abr,old} - B_{cbr}[VC] - B_{vbr}[VC] \quad (9)$$

$$\lambda_{abr,new} = \lambda_{abr,old} + B_{abr}[VC] / AI \quad (10)$$

한편 현재 연결된 특정 VC가 순방향으로 스위치에서 연결 해제를 완료하면 EFRA는 이용중인 대역폭을 다른 VC들에게 즉시 분배한다. EFRA는 연결 해제 후에 다음 AI 시간에 식 (11) ~ (13)의 변수를 새로 계산한다. ABR 연결의 수는 식 (11)과 같이 하나가 감소하고, 새로운 서비스가 CBR, VBR이면 식 (1)의 가용 ABR 대역폭은 식 (12)와 같이 가산된다. 반면 새로운 서비스가 ABR이면 ABR 트래픽 유입률은 식 (13)과 같이 즉시 감소된다.

$$C_{abr,new} = C_{abr,old} - 1 \quad (11)$$

$$\text{가용 } B_{abr,new} = \text{가용 } B_{abr,old} + B_{cbr}[VC] + B_{vbr}[VC] \quad (12)$$

$$\lambda_{abr,new} = \lambda_{abr,old} - B_{abr}[VC] / AI \quad (13)$$

### 3.4 EFRA의 문제점

EFRA는 ERICA+의 장점인 정확한 공정률 할당,

100%의 링크 이용률, 낮은 수행 복잡도 등을 계승하고, 연결 설정 과정에서 발생하는 트래픽 협상 정보를 조기에 공정률 계산 과정에 반영하여 알고리즘의 성능을 과도에서 정상 상태로 신속하게 회복시키는 장점이 있다. 그리하여 EFRA는 ERICA+보다 버퍼의 큐 길이가 최대 및 평균값에서 상당히 낮아졌고, 전송 처리 지연과 전송 처리율, 보다 정확한 공정률의 할당 등에서 성능이 향상되었다. 그러나 EFRA와 ERICA+는 종종 부정확하게 최대-최소 공정률을 할당하고, 출력 링크의 대역폭을 최대한 100%까지 이용하기 위하여 가용 ABR 대역폭을 높게 계산함으로써 결국 버퍼에는 ABR 데이터가 지나치게 많이 저장된다. ABR 서비스는 트래픽의 전송 특성을 예측하기 힘들기 때문에 100%의 링크 이용률을 유지하려면 스위치의 버퍼에는 일정한 ABR 데이터를 저장해야 한다. 그러나 버퍼내의 많은 데이터는 전송 처리 지연과 낮은 전송 처리율을 발생시키고, 유한한 버퍼의 크기는 데이터가 손실될 확률을 증가시키고, 송신단으로 RM 셀의 전달 지연을 가져와 정확한 최대-최소 공정률의 계산에 오차를 가져온다. 그리하여 망의 혼잡은 장시간 지속되어 스위치 버퍼의 큐 길이는 매우 천천히 감소하고, 과도에서 정상 상태로 느리게 천이하여 장시간 성능 감쇠가 지속된다. 그리고 망의 자원이 급격하게 변동하면 최대-최소 공정률은 느리게 할당되고, 버퍼의 ABR 데이터는 급격히 증가하여 알고리즘은 다양한 성능 지표 부문에서 성능 감쇠가 나타난다. 또한 EFRA는 초기 세대의 스위치에서 사용하는 혼잡 제어 방법을 지원하지 않기 때문에 명시적 전송률 제어에 의한 혼잡 제어의 장점을 최대한 활용할 수가 없다.

## IV. 제안 EFRA+ 알고리즘

EFRA+는 ERICA+와 EFRA의 주요 장점의 계승과 동작의 단점을 보완하고, 연결 제어 정보의 조기 참조, 보다 정확한 최대-최소 공정률의 할당, 과도에서 정상 상태로의 신속한 천이, 그리고 초기 세대의 스위치와 상호 동작 등의 부문에서 성능을 개선한 알고리즘이다. EFRA+는 EFRA의 연결 제어 정보의 반영 과정과 최대-최소 공정률의 할당 과정을 수정함으로써 수많은 연결의 설정 과정에서 발생하는 망의 잠재적인 혼잡을 예방하고, 과도 상태에서 발생하는 성능 감쇠를 크게 향상시키는데 목표가 있다. 그리하여 버퍼의 데이터는 낮은 범위에서 관리

하여 빠르게 데이터를 전송 처리함으로써 망의 스위치들이 공정을 신속하게 공유하여서 망 전체의 혼잡을 효율적으로 제어할 수 있도록 한다. 또한 초기의 binary 스위치와 상호 연동할 수 있도록 초기의 혼잡 제어 방법을 병행하여 수행함으로써 명시적 전송률 할당 알고리즘의 성능 장점을 최대화할 수 있도록 한다. 다음은 EFRA 알고리즘의 주요 동작 과정이다.

4.1 공정 할당값의 계산

스위치는 주기적인 AI 시간 간격마다 CBR과 VBR 서비스의 점유 대역폭, 총 ABR 대역폭, 가용 ABR 대역폭, 그리고 출력 링크별로 모든 ABR 서비스의 VC들이 공평하게 서로 공유하는 공정 할당값을 계산한다. 공정 할당값의 계산 수식은 식 (16) 과 같다. 한편 EFRA+는 EFRA와 달리 부하 인자(z), ABR 유입률( $\lambda_{abr}$ )은 사용하지 않는다.

$$B_{total\_abr} = B_{link} - B_{total\_signal} - B_{cbr\_vbr} \tag{14}$$

$$B_{available\_abr} = B_{total\_abr} \times F(T_q) \tag{15}$$

$$\gamma_i = \frac{B_{available\_abr} \cdot e_{abr}}{N_{total\_abr}} \tag{16}$$

여기에서,  $B_{link}$ 는 스위치에서 각 출력 포트의 링크 대역폭이고,  $B_{cbr\_vbr}$ 는 스위치로 유입되는 CBR과 VBR 서비스의 점유 대역폭으로서 출력 포트별로 직접 측정하여 계산한다.  $B_{total\_signal}$ 은 연결 설정 과정에서 EFRA+가 조기에 참조하는 대역폭의 합계이다.  $B_{total\_abr}$ 는 ABR 서비스가 사용하는 총 ABR 대역폭이다. 이 값은 EFRA와 달리 연결 제어 과정에서 발생한 트래픽 파라미터의 대역폭 값을 반영하여 계산한다. 식 (15)의  $B_{available\_abr}$ 는 가용 ABR 대역폭이며, 큐잉 지연과 버퍼의 큐 길이와 관련된 큐 제어 함수  $F(T_q)$ 를 이용하여 계산한다. 그림 2는 ERICA+, EFRA, EFRA 알고리즘의 함수  $F(T_q)$ 이다.

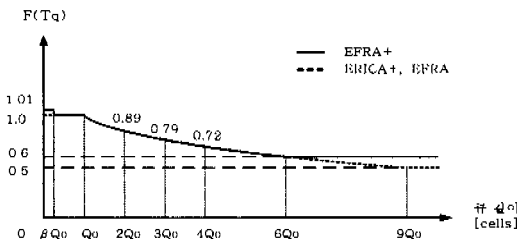


그림 2. 3가지 알고리즘의 함수  $F(T_q)$ 의 비교

식 (17)에서  $T_0$ 은 큐잉 지연 시간을 나타내는 상수이다.  $T_0$ 은 버퍼에 적정한 ABR 데이터  $Q_0$ 가 저장되도록 설정해야 하며, 제안하는 값은 LAN에서는  $50 \sim 100\mu sec$ , WAN에서는  $300 \sim 400\mu sec$ 이다.  $Q_0$ 은 큐잉 지연 시간과 총 ABR 대역폭을 승산하여 계산한다. 그림 2에서 큐 제어 함수  $F(T_q)$ 는 버퍼의 큐 길이 범위에 따라 최소 0.6에서 최대 1.01 범위의 상수값을 가진다. 함수  $F(T_q)$ 는 큐 길이가  $\beta Q_0$  이하이면 상수 b에 해당하는 최대값을 가진다. 상수 b의 제안값은 1.01 이고, 버퍼의 큐 길이를 일정한 크기만큼 증가시킨다. 상수  $\beta$ 는 0에서 1 사이의 값이고, 제안값은 0.1 이다. 큐 길이가  $6Q_0$ 을 초과하면 함수  $F(T_q)$  값은 최소값인 0.6으로 일정하다. 다른 큐 길이 구간에서는 3가지 알고리즘은 동일한 큐 제어 함수값을 사용한다.

$$Q_0 = B_{total\_abr} \times T_0 \tag{17}$$

$$F(T_q) = \frac{aQ_0}{(a-1)Q + Q_0} \tag{18}$$

$$\left[ \begin{array}{ll} "0 \leq Q \leq \beta Q_0", & "F(T_q) = b" \\ "\beta Q_0 < Q \leq Q_0", & "F(T_q) = 1.0" \\ "Q_0 < Q \leq 6Q_0", & "a = 1.15" \\ "6Q_0 < Q", & "F(T_q) = 0.6" \end{array} \right]$$

4.2 새로운 공정률 할당 방식

스위치에 i 번째 VC의 BRM 셀이 도착하면 조정된 공정 할당값( $\gamma_i$ )과 각 VC별로 최대-최소 공정률을 계산하여 BRM에 새로운  $ER_i$  값을 대체하여 이웃 스위치로 전송한다. 먼저 식 (19)에서 연결 설정 과정에서 초기 반영하였던 최소 전송률은 연결 제어 대역폭에서 감소되고, 가용 ABR 대역폭은 최소 전송률만큼 증가한다. 또한 총 ABR VC의 수는 하나가 증가하고 조정된 공정 할당값은 식 (20)에 의하여 즉시 새로 계산된다.  $B_{total\_signal}$ 과  $B_{i\_abr\_MCR}$ 은 조기에 참조한 연결 제어 정보로서 4.3 절에서 설명한다.

$$\left. \begin{array}{l} \text{If (첫 번째 BRM}_i \text{)} \{ \\ B_{total\_signal} = B_{total\_signal} - B_{i\_abr\_MCR} \\ N_{total\_abr} = N_{total\_abr} + 1 \\ B_{available\_abr} = B_{available\_abr} + B_{i\_abr\_MCR} \\ \} \end{array} \right\} \tag{19}$$

$$\gamma_i = \frac{B_{available\_abr} \cdot e_{abr}}{N_{total\_abr}} \tag{20}$$

최대-최소 공정률은 조정된 공정 할당값으로 계산한다. 먼저 식 (21)처럼 VC의 최대 전송률이 공정 할당값보다 작으면 배분된 공정 할당값은 모두 사용하지 못한다. 그러므로 최대-최소 공정률은 최대 전송률로 할당해 주고, 공정률과 최대 전송률의 차이값은 가용한 다른 VC들에게 식 (22)에 의하여 배분된다.  $N_{total\_bottlenecked}$ 는 병목 현상이 발생한 VC들의 총 수이고,  $B_{total\_PCR\_bottlenecked}$ 는 병목된 VC들의 최대 전송률 합계로서 출력 포트별로 계산한다.  $S_{i,bottlenecked}$ 는 각 VC별로 병목이 발생했는지를 나타내는 1 비트 변수이고, BRM 셀이 도착할 때마다 병목 여부가 판단되어 처리된다.

$$\begin{aligned}
 & \text{If } (\gamma_i > B_{i,abr\_PCR}) \{ \\
 & \quad \text{If } (S_{i,bottlenecked} \neq \text{TRUE}) \{ \\
 & \quad \quad S_{i,bottlenecked} = \text{TRUE} \\
 & \quad \quad N_{total\_bottlenecked} = N_{total\_bottlenecked} + 1 \\
 & \quad \quad B_{total\_PCR\_bottlenecked} = B_{total\_PCR\_bottlenecked} + B_{i,abr\_PCR} \\
 & \quad \} \\
 & \quad ER_i = B_{i,abr\_PCR} \quad (21) \\
 & \}
 \end{aligned}$$

$$\begin{aligned}
 & \text{Else } \{ \\
 & \quad \text{If } (S_{i,bottlenecked} == \text{TRUE}) \{ \\
 & \quad \quad S_{i,bottlenecked} = \text{FALSE} \\
 & \quad \quad N_{total\_bottlenecked} = N_{total\_bottlenecked} - 1 \\
 & \quad \quad B_{total\_PCR\_bottlenecked} = B_{total\_PCR\_bottlenecked} - B_{i,abr\_PCR} \\
 & \quad \} \\
 & \quad ER_i = \gamma_i + \\
 & \quad \frac{N_{total\_available\_d} \cdot \gamma_i - B_{total\_available\_d}}{N_{total\_available\_d} - N_{total\_bottlenecked\_d}} \quad (22) \\
 & \}
 \end{aligned}$$

최종적으로 EFRA+는 초기 세대의 스위치와 혼잡 제어의 상호 연동을 위하여 버퍼의 큐 길이에 따라 혼잡 판단 정보를 식 (23)과 같이 BRM 셀의 해당 필드에 처리하고, 역방향 RM 셀 내의 ER 필드는 식 (24)과 같이 새로운 공정률로 대체된다.

$$\begin{aligned}
 & \text{If } (QT < Q \leq DQT) \quad Cl_{i,BRM} = 1 \\
 & \text{Else if } (DQT < Q) \quad Cl_{i,BRM} = N_{i,BRM} = 1 \quad (23)
 \end{aligned}$$

$$ER_{i,BRM} = \min\{ER_{i,BRM}, \min\{B_{available\_abr}, ER_i\}\} \quad (24)$$

### 4.3 연결 제어 정보의 참조 개선

새로운 i 번째 VC가 스위치에서 역방향으로 연

결 설정을 완료하면 각 VC별로 연결 정보 테이블은 생성된다. 이때 EFRA+는 VC와 관련된 트래픽 파라미터인 대역폭(B)을 서비스에 따라 식 (25) ~ (27)과 같이 저장한다. CBR은 PCR 값, VBR은 SCR 값, 그리고 ABR은 MCR과 PCR 값을 이용한다. 이들 대역폭은 공정 할당값의 계산에서 참조하기 위하여 출력 포트별로 연결 대역폭의 합계 ( $B_{total\_signal}$ ) 변수에 가산된다. 이 합계 값은 총 ABR 대역폭을 계산할 때 이용하고, 보다 정확한 공정 할당값의 계산에서 중요하게 참조된다.

$$\begin{aligned}
 & \text{If (역방향으로 새로운 연결 설정)} \\
 & \quad \text{If (CBR 서비스) } \{ \\
 & \quad \quad B_{i,cb\_PCR} = \text{PCR} \\
 & \quad \quad B_{total\_signal} = B_{total\_signal} + B_{i,cb\_PCR} \quad (25) \\
 & \quad \} \\
 & \quad \text{Else if (VBR 서비스) } \{ \\
 & \quad \quad B_{i,vbr\_SCR} = \text{SCR} \\
 & \quad \quad B_{total\_signal} = B_{total\_signal} + B_{i,vbr\_SCR} \quad (26) \\
 & \quad \} \\
 & \quad \text{Else if (ABR 서비스) } \{ \\
 & \quad \quad B_{i,abr\_MCR} = \max(\text{ICR}, \text{MCR}) \\
 & \quad \quad B_{i,abr\_PCR} = \text{PCR} \\
 & \quad \quad B_{total\_signal} = B_{total\_signal} + B_{i,abr\_PCR} \quad (27) \\
 & \quad \}
 \end{aligned}$$

연결 설정 과정에서 저장한 식 (25) ~ (27)의 대역폭은 한정된 시간동안 참조된다. 만약 새로운 VC가 CBR, VBR 서비스이면 연결 설정을 완료한 후부터 송신단이 첫 번째 데이터를 전송하고, 그 데이터가 해당 스위치에 도착할 때까지 대역폭은 계속 이용된다. 첫 번째 데이터가 스위치에 도착한 직후에는 식 (25), (26)의 대역폭은 삭제되고, 출력 포트별로 연결 대역폭의 합계는 식 (28), (29)를 이용하여 각 VC의 대역폭만큼 감소된다. 하지만 ABR 서비스이면 식 (27)의 대역폭은 연결 해제시까지 참조한다.

$$\begin{aligned}
 & \text{If (각 VC의 첫번째 데이터 셀)} \\
 & \quad \text{If (CBR 서비스)} \\
 & \quad \quad B_{total\_signal} = B_{total\_signal} - B_{i,cb\_PCR} \quad (28)
 \end{aligned}$$

$$\begin{aligned}
 & \quad \text{Else if (VBR 서비스)} \\
 & \quad \quad B_{total\_signal} = B_{total\_signal} - B_{i,vbr\_SCR} \quad (29)
 \end{aligned}$$

다음으로 현재 전송중인 ABR VC가 연결을 해제하면 식 (30)처럼 총 ABR VC들의 수는 하나가

감소한다. 만일 VC가 병목 현상이 발생한 상태였다면 식 (31)처럼 출력 포트별로 병목이 발생한 총 ABR VC들의 수는 하나가 감소하고, 연결 대역폭의 합계는 최대 전송률만큼 감소된다.

$$N_{total\_abr} = N_{total\_abr} - 1 \quad (30)$$

If ( $S_i$ , bottlenecked == TRUE) {

$$N_{total\_bottlenecked} = N_{total\_bottlenecked} - 1$$

$$B_{total\_PCR\_bottlenecked} = B_{total\_PCR\_bottlenecked} - B_{i,abr\_PCR} \quad (31)$$

}

### V. 실험 결과

컴퓨터 시뮬레이션은 NIST(National Institute of Standards and Technology)에서 개발하여 배포한 "NIST ATM/HFC Network Simulator Ver 4.0"을 이용하였으며, ATM 연결 설정과 해제 메커니즘은 프로그램을 수정하여 추가하였다.

#### 5.1 실험 환경 설정

그림 3~6은 실험을 위한 네트워크 구성도이다. 각 링크 구간 거리는 300 km, 대역폭은 155 Mbps, 그리고 전파 지연은 1 msec이다. ATM 스위치는 비블로킹(non-blocking), 출력 버퍼형인 명시율 기반 스위치이고, 서비스 부류별로 FIFO 방식으로 큐잉을 하며, 계층적 순환 방식으로 셀 스케줄링을 한다. 표 2는 실험에서 공통으로 사용하는 ABR 파라미터이고, 표 3은 각 알고리즘의 파라미터이다.

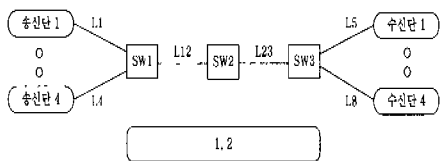


그림 3. 단순한 망 구성도

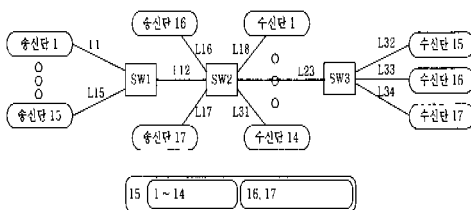


그림 4. 상향 흐름의 망 구성도

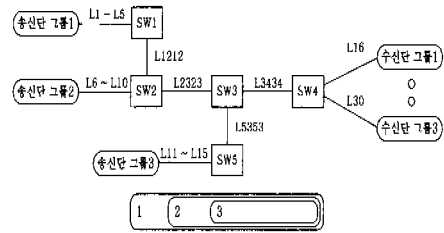


그림 5. 다중 경로의 망 구성도

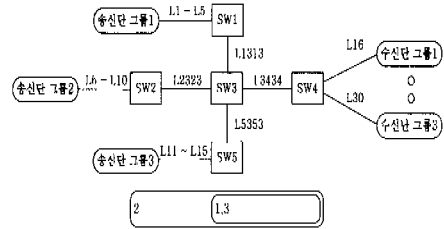


그림 6. 하향 흐름의 망 구성도

표 2. 송신단의 ABR 파라미터

파라미터	설정값	파라미터	설정값
PCR	155 Mbps	TBE	160,000
ICR	MCR, PCR/20	Nrm	32
RIF	1	Trm	100 msec
RDF, CDF	1/16	ADTF	500 msec

표 3. 각 알고리즘의 파라미터

파라미터	ERICA+	EFRA	EFRA+
$T_0$	400 $\mu$ sec	300 $\mu$ sec	300 $\mu$ sec
$F_{min}$	0.5	0.5	0.6
a	1.15	1.15	1.15
b	1.00	1.00	1.01
AI	400 $\mu$ sec	300 $\mu$ sec	300 $\mu$ sec
$\alpha$	0.8	0.8	해당없음
$\delta$	0.1	0.1	해당없음
$\beta$	해당없음	해당없음	0.1
Decay Factor	0.9	해당없음	해당없음
QT	해당없음	해당없음	2 $Q_0$ cells
DQT	해당없음	해당없음	3 $Q_0$ cells

표 4~7은 각 실험에서 설정한 트래픽 파라미터이다. 표 4와 5의 VBR 서비스는 균집과 휴지 시간이 각각 10-5와 5-5 msec인 ON-OFF 트래픽이다. 표 6과 7의 각 송신단 그룹은 4개의 ABR과 1개의 VBR 서비스로 구성되어 있고, ABR 서비스가 먼저 전송 출발한다. 하향 흐름 망에서 스위치 SW3은

ER 스위치, 나머지는 binary 스위치이다. binary 스위치의 큐 길이 QT와 DQT는 1000, 2000 cells이다. 모든 실험에서 각 송신단들은 일정한 전송 출발 지연을 두고 연결 설정을 하고, 수신단과 연결 설정을 완료한 후에 실제 데이터를 전송하기 시작한다.

표 4. 단순한 망의 트래픽 파라미터

송신단	서비스	출발시간	총전송시간	전송속도
송신단1	ABR	0	160	MCR = PCR/20
송신단2		20	140	
송신단3		40	120	
송신단4	VBR	60	100	SCR= 50

표 5. 상향 흐름 망의 트래픽 파라미터

송신단	서비스	출발시간	전송속도
송신단1	VBR	20	SCR= 50
송신단15	ABR	0	MCR = PCR/20
송신단2~17	ABR	5 (각 간격은 5)	

표 6. 다중 경로 망의 트래픽 파라미터

송신단	출발시간	VBR트래픽	전송속도
그룹1	0 (각 간격은 5)	5-5 msec	SCR= 20, MCR = PCR/30
그룹2	25 (각 간격은 5)	5-10 msec	
그룹3	50 (각 간격은 5)	5-15 msec	

표 7. 하향 흐름 망의 트래픽 파라미터

송신단	출발시간	VBR트래픽	전송속도
그룹1	0 (각 간격은 10)	5-5 msec	SCR= 20, MCR = PCR/30
그룹2	50 (각 간격은 10)	5-10 msec	
그룹3	50 (각 간격은 10)	5-15 msec	

5.2 실험 결과

그림 7~10은 각 실험에서 측정한 송신단1(ABR 서비스)의 현재 전송률의 변화이다. 3가지 알고리즘 중에서 EFRA+가 EFRA, ERICA+보다 정상 상태에서는 매우 안정적으로 동작하고, 과도 상태에서는 신속하게 공정률을 재분배한다. 즉, 새로운 연결 설정으로 망의 자원이 급격하게 변하는 과도 상태에서는 연결 설정 정보를 조기에 공정률의 할당 과정에 반영하였기 때문에 공정률은 신속, 정확하게 새로운 최적의 값으로 분배되어서 스위치에서의 혼잡은 미리 예방된다. 그리고 균집성이 강하여 트래픽

의 전송 특성이 매우 변동이 심한 VBR 서비스에 의하여 발생하는 과도 상태에서도 신속하게 반응하여 응답한다. 그 결과 스위치 버퍼의 ABR 데이터는 낮은 범위에서 유지되어 스위치의 전송 처리 지연은 감소한다. 또한 RM 셀에 의한 피드백 정보의 전송 지연도 감소시켜서 모든 송신단과 스위치들이 최대-최소 공정률을 신속하게 참조함으로써 향후에 계산된 공정률은 보다 정확하게 할당된다.

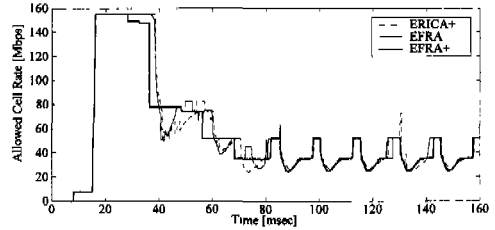


그림 7. 단순한 망에서 ABR1의 전송률

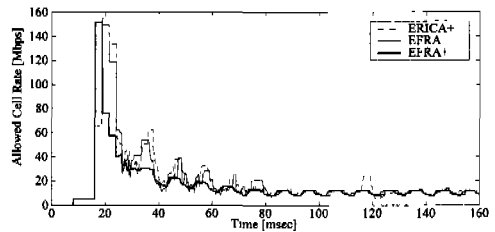


그림 8. 상향 흐름 망에서 ABR15의 전송률

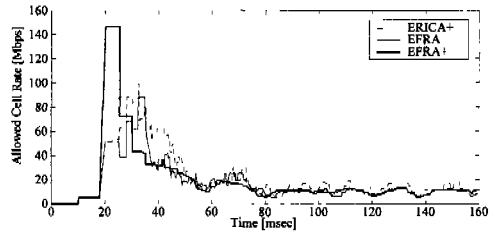


그림 9. 다중경로 망에서 ABR1의 전송률

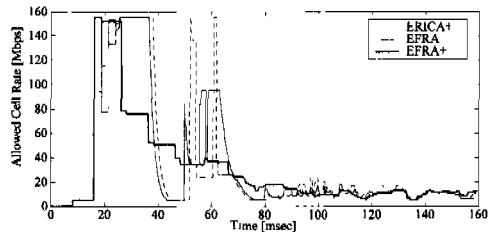


그림 10. 하향 경로 망에서 ABR1의 전송률



표 8과 그림 11, 12는 혼잡이 발생한 ER 스위치에서 측정된 버퍼의 큐 길이다. EFRA+를 기준으로 EFRA와 ERICA+의 큐 길이를 배수화하여 표시하였다. EFRA+는 ERICA+, EFRA보다 낮은 범위에서 관리하여 다양한 성능 지표들을 최대한 만족시키는 것으로 평가되었다. EFRA+는 연결 제어 정보의 초기 반영과 보다 정확한 최대-최소 공정률의 할당으로 인하여 버퍼의 ABR 큐 길이는 EFRA보다 최대값은 40 ~ 66%, 평균값은 30 ~ 70%씩 감소하였다. 이러한 낮은 ABR 큐 길이 때문에 스위치에서 데이터의 전송 처리 지연도 ABR 큐 길이와 동일한 비율로 감소하였다. 한편 ERICA+와 비교하면 최대값은 50 ~ 70%, 평균값은 50 ~ 70%씩 감소하였다. EFRA와 ERICA+는 버퍼에 지나치게 많은 ABR 데이터를 저장하였고, 다양한 성능 지표들의 성능을 감쇠시키는 문제점이 발생하였다.

표 8. 스위치에서 ABR 큐 길이의 실험 결과 [단위: cells]

실험	스위치	EFRA+		EFRA		ERICA+	
		최대	평균	최대	평균	최대	평균
실험1	SW1	315	82	576	157	639	216
실험2	SW1	325	143	631	210	1097	293
실험3	SW3	547	150	791	271	1118	346
실험4	SW3	624	148	1725	472	1870	480

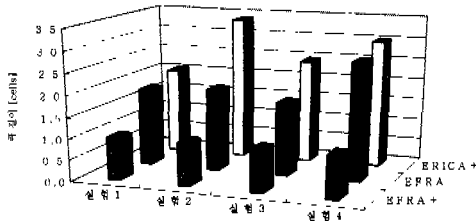


그림 11. 최대 큐 길이의 성능 비교

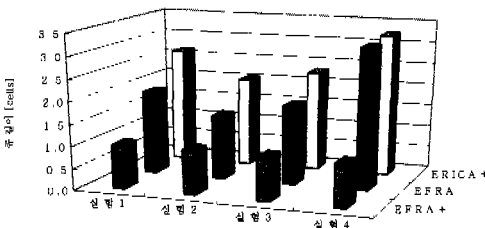


그림 12. 평균 큐 길이의 성능 비교

표 9와 그림 13은 이상적인 경우를 기준으로 EFRA+, EFRA, ERICA+의 출력 링크 이용률을 배수화하여 표시하였다. 명시적 공정률 할당 알고리즘은 출력 링크의 이용률을 최대한 100%까지 유지해야 한다. 일반적으로 높은 링크 이용률보다는 버퍼의 ABR 큐 길이를 낮게 유지하는 것이 더 중요하다. 4개의 실험에서 링크 이용률의 평균값은 EFRA+, EFRA, ERICA+이 각각 0.92, 0.93, 1.0 이고, 이용률의 범위는 0.90 ~ 0.94, 0.87 ~ 1.0, 0.87 ~ 1.16으로 측정되었다. EFRA+는 일정한 범위에서 안정적으로 이용하였고, EFRA, ERICA+는 다소 변동이 높았다. ERICA+는 이용률이 2개의 실험에서 1.0을 초과하였는데, 부정확한 공정률의 할당으로 버퍼에 ABR 데이터가 지나치게 많이 저장되었기 때문이다.

표 9. 링크 이용률의 실험 결과 [단위: msec]

실험	링크	Ideal	EFRA+	EFRA	ERICA+
실험1	L12	130.0	117.9	118.7	113.1
실험2	L1212	116.0	109.3	115.5	134.1
실험3	L3434	119.0	107.6	103.0	112.5
실험4	L3434	124.0	112.9	118.2	128.4

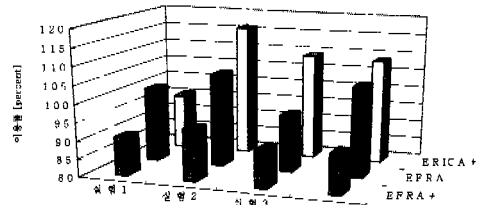


그림 13. 링크 이용률의 성능 비교

표 10과 그림 14는 binary와 ER 스위치 버퍼의 ABR 큐 길이를 통계한 데이터이다. 실험의 초기 시간에는 데이터가 binary 스위치 SW1을 경유하여 전송된다. 이때 ER 스위치 SW3은 공정률을 분배하여 송신단들에게 전달한다. EFRA와 ERICA+의 경우는 각 송신단들이 가장 가까이 위치한 binary 스위치의 혼잡 제어에 더 영향을 받기 때문에 스위치 SW1의 ABR 큐 길이는 EFRA+보다 최대 및 평균 값이 무려 18배나 많게 유지된다. 그러나 EFRA+는 ER 스위치 SW3이 분배한 공정률에 따라 혼잡이 제어되어 스위치 SW1의 큐 길이는 관리하고자하는

범위에서 유지되고 있다. EFRA와 ERICA+는 binary 스위치의 혼잡 제어 방법을 수용하지 않기 때문에 공정률은 매우 부정확하게 할당되고, 버퍼에는 매우 많은 ABR 데이터가 저장되었다. EFRA+는 binary 스위치의 혼잡 제어 방법을 수용함으로써 스위치들의 구분없이 안정적으로 최대-최소 공정률을 할당하는 것으로 평가되었다.

표 10. binary 스위치의 ABR 큐 길이 [단위: cells]

실험	스위치	EFRA+		EFRA		ERICA+	
		최대	평균	최대	평균	최대	평균
실험4	SW1	181	82	3261	1580	3109	1558
	SW2	1	1	1	1	1	1
	SW3	624	148	1725	472	1870	480
	SW5	1	1	1	1	1	1

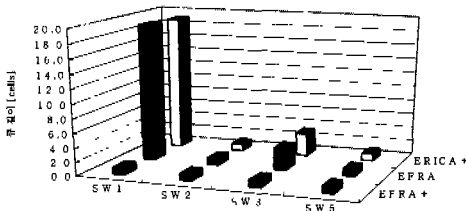


그림 14. binary 스위치의 ABR 큐 길이

## VI. 결론

본 논문에서는 연결 설정 동안 발생한 연결 제어 정보를 이용하는 향상된 조기 공정률 할당 알고리즘인 EFRA+를 제안하였다. EFRA+는 EFRA의 주요 특징인 연결 제어 정보의 반영, O(1)의 낮은 계산 복잡도, 높은 출력 링크 이용률 등의 주요 장점을 계승하였다. 그리고 스위치에서의 잠재적인 혼잡을 방지하기 위하여 보다 신속하게 연결 제어 정보를 공정률의 할당 과정에 조기에 반영하여 과도 상태에서의 성능 열화를 개선하고, 보다 정확한 최대-최소 공정률을 계산하는 것에 목표를 두었다. 실험 결과와 분석을 종합하면 EFRA+는 혼잡 제어의 수행 복잡도가 낮으면서도 보다 정확하게 최대-최소 공정률을 할당하였다. 그 결과 정상 상태에서는 안정적으로 공정률을 할당하였고, 연결 제어 정보를 조기에 참조하여 과도 상태에서는 신속하게 정상 상태로 천이하였다. 또한 다양한 VBR 트래픽과의

적용에서도 신속하게 대역폭의 변동을 반영하여 빠른 과도 상태 응답을 보였다. 스위치 버퍼에는 낮은 범위에서 ABR 데이터를 저장하면서도 출력 링크의 이용률은 매우 높게 유지하였다. 버퍼의 ABR 큐 길이는 EFRA보다 최대값은 40 ~ 66%, 평균값은 30 ~ 70%씩 감소하였다. 또한 데이터의 전송 처리 지연도 ABR 큐 길이와 동일한 수치로 감소하였다. 그리고 초기의 혼잡 제어 방법을 병행함으로써 binary 스위치와 상호 연동하여 혼잡과 흐름을 제어할 때에도 효율적으로 동작하는 것으로 평가되었다.

## 참고 문헌

- [1] ATM Forum Technical Committee, "Traffic Management Specification Version 4.1", ATM Forum Contribution 99-0121.000, Mar. 1999.
- [2] 김대일, 이진우, 임현주, 김남일, "ATM 망에서 ABR 흐름 제어의 과도 상태의 성능 개선", 한국통신학회 논문지 제 25권 제 6T호, pp. 25-33, Jun. 2000.
- [3] 김남일, "ATM 망에서 신호 메커니즘을 이용한 조기 공정률 할당 알고리즘", 전국대학교 박사학위 논문, Jun. 2000.
- [4] 김대일, 임현주, 박인갑, "ATM 망에서 연결제어 메커니즘을 이용한 혼잡 회피 EFRA 알고리즘", 전기전자학회 논문지 제4권 제 1호, pp. 67-76, Jul. 2000.
- [5] R. Jain, S. Kalyanaraman, R. Goyal, S. Fahmy, and R. Viswanathan, "ERICA Switch Algorithm: A Complete Description", ATM Forum Contribution 96-1172, Aug. 1996.
- [6] R. Jain, S. Fahmy, S. Kalyanaraman, and R. Goyal, "ABR Switch Algorithm Testing: A Case Study with ERICA", ATM Forum Contribution 96-1267, Oct. 1996.
- [7] William Stallings, High-Speed Networks: TCP/IP and ATM Design Principles, Prentice Hall, pp.344-376, Sep. 1997.
- [8] ATM Forum Technical Committee, "ATM User-Network Interface Signaling Specification Version 4.0", ATM Forum Contribution 96-0061.000, Jul. 1996.
- [9] A. Arulambalam, X. Chen and N. Ansari, "Allocating Fair Rates for Available Bit Rate Service in ATM Networks", IEEE Commun.

- Magazine, vol. 34, pp. 92-100, Nov. 1996.
- [10] A. Arulambalam, X. Chen and N. Ansari, "An Intelligent Explicit Rate Control Algorithm for Service in ATM Networks", Proc. IEEE ICC'97, vol 1/3, Jul. 1997.
- [11] Uyless Black, ATM Volume II: Signaling in Broadband Networks, Prentice Hall, pp. 92-179, Aug. 1997.
- [12] Nada Golmie, et al., "The NIST ATM/HFC Network Simulator Operation and Programming Guide Version 4.0", NIST Information Tech. Lab., Dec. 1998.

김 대 일(Dae-Il Kim)                                      준회원  
1992년 2월 : 건국대학교 전자공학과(공학사)  
1995년 8월 : 건국대학교 전자공학과(공학석사)  
2001년 2월 : 건국대학교 전자공학과(박사예정)  
<주관심 분야> ATM 트래픽 제어, 큐잉이론, ATM 망 접속 기술

박 기 태(Ki-Tae Park)                                      준회원  
1993년 2월 : 원광대학교 전자공학과(공학사)  
1996년 8월 : 건국대학교 전자공학과(공학석사)  
1999년 8월 : 건국대학교 전자공학과(박사수료)  
<주관심 분야> ATM 트래픽 제어, 큐잉이론, 컴퓨터 네트워크

김 남 일(Nam-Il Kim)                                      정회원  
한국통신학회 논문지 제24권 11T호 참조  
1999년 3월 ~ 현재 : 가천길대학 전자통신과 조교수