

ATM망에서 ABR서비스를 위한 효율적인 전송률 할당 알고리즘에 관한 연구

천재호 · 윤상호 · 김명균 · 허정석
컴퓨터 · 정보통신 공학부

<요 약>

ATM망에서 ABR 서비스는 CBR, VBR에 할당된 대역폭을 제외한 나머지를 이용해서 데이터를 전송한다. 오디오 또는 비디오 데이터와 같이 실시간 전송이 필요한 서비스들을 전송지연이 제한되는 CBR이나 VBR 트래픽을 이용하여 제공할 수 있지만, 이런 경우에는 전송 대역폭을 효율적으로 사용하지 못하는 단점이 있다. 본 논문에서는 실시간 데이터를 ABR 서비스를 이용하여 효율적으로 전송하기 위한 ABR 대역폭 할당 알고리즘을 제안한다. 제안된 알고리즘은 MCR을 보장하며, MCR에 비례해 전송률을 할당한다. 그리고 접속의 상태를 만족(satisfied) 상태와 불만족(bottlenecked) 상태로 구분하고, 만족 상태인 접속이 사용하지 않는 대역폭을 불만족 상태인 접속이 사용할 수 있도록 하여 대역폭 활용도를 높인다. 또한 제안된 알고리즘은 큐 제어 함수를 사용해서 큐 길이를 일정한 범위로 유지함으로써 셀의 전송지연을 일정하게 한다. 본 논문에서는 시뮬레이션을 통해 제안된 알고리즘의 성능을 분석하고, 기존에 제안된 ATM 포럼[1] 및 Kalampoukas 알고리즘[2]과 비교를 하였다.

A Study on the Efficient Transmission Rate Allocation Algorithm for ABR Services in ATM Networks

Jae-Ho Chun · Sang-Ho Yoon · Myung-Kyun Kim
Joung-Seuk Heo
School of Computer and Information Communication

<Abstract>

On ATM networks, ABR services are provided using the remained bandwidth after allocating CBR and VBR traffic. Realtime services such as transmitting audio or video data may be provided using CBR and VBR which have a constrained transmission delay, but in those cases, the communications bandwidth may be wasted. This paper proposes an efficient ABR rate allocation algorithm to transfer real-time data using ABR service. The proposing algorithm guarantees MCR and allocates bandwidth to each connection proportional to its MCR. The proposing algorithm divides the connections in two groups, a satisfied state group and a bottlenecked state group, and enhances bandwidth utilization by allowing the remained bandwidth after allocating to the connections in the satisfied state to the connections in the bottlenecked state. The proposing algorithm uses the queue control function to keep the queue length within some boundary, which makes the transmission delay constant. We simulate and compare the performance of the proposing algorithm with that of the algorithms proposed by ATM Forum[1] and Kalampoukas[2].

1. 서론

미래의 고속통신망에서는 정보를 전송하기 위해 고정 크기의 셀을 사용하는 비동기 전송 방식(ATM)을 사용할 것이다. ATM은 다양한 서비스들을 효과적으로 지원할 수 있으므로 미래의 망 구현 측면에서 가장 유망한 기술이 되었다[2]. ATM 포럼은 사용자의 다양한 요구를 만족시키기 위해 트래픽의 특성과 서비스 품질에 따라 5개의 서비스로 분류하였다 [1]. 서비스는 크게 실시간과 비 실시간 서비스로 구분된다. 실시간 서비스는 CBR(Constant Bit Rate), rt-VBR(real time Variable Bit Rate)로 구성되며 비 실시간 서비스는 nrt-VBR(non-real time Variable Bit Rate), ABR(Available Bit Rate), UBR(Unspecified Bit Rate)로 구성된다. ABR 서비스는 CBR과 VBR 서비스에 할당하고 남은 나머지 대역폭을 사용해서 데이터를 전송한다. 전송률을 할당하는 알고리즘은 크게 사용권 기반 방식(Credit Based Scheme)과 전송률 기반 방식(Rate Based Scheme)으로 나누어지며 ATM 포럼에서는 전송률 할당 알고리즘으로 전송률 기반방식을 채택하였다. 전송률 기반방식은 소스에서 망의 상태에 대한 정보를 얻기 위해서 일정한 수의 데이터 셀마다 RM(Resource Management)셀을 전송한다. 스위치를 거쳐 테스트네이션에 도착한 RM셀은 소스 측으로 다시 되돌려지며 소스는 가장 혼잡이 심한 스위치에서 할당된 전송률로 소스의 전송률을 조정하게 된다. 이러한 전송률 할당 알고리즘의 목적은 높은 대역 활용도, 작은 큐 지연, 작은 셀 손실을 유지하고 경쟁하는 소스들 사이에서 공정성을 보장하는 것이다.

ABR 서비스를 이용하여 실시간 데이터를 전송하기 위한 연구는 Kalampoukas와 ATM Forum 등에서 많이 연구되었다 [2, 3, 4, 5, 6]. Kalampoukas 논문에서 제시한 알고리즘

의 수행 시간은 $O(1)$ 이며, 이는 Charny에 의해 제안된 MIT Scheme의 $O(n)$ 의 수행시간과 유사하다[2, 4]. 총 가능 용량(total available capacity)이나 접속의 요구량이 변할 때마다 최대-최소 할당(Max-min allocation)에 따라 전송률을 할당한다[4]. 이 알고리즘은 접속들간의 상태를 만족 상태와 불만족 상태로 나누어 각기 다른 알고리즘을 수행한다. 수행하는 알고리즘은 아래와 같다.

$$A_{\max}(t) = B_{eq}(t) + \frac{B_f(t)}{N_b(t)} \quad (\text{불만족 상태})$$

$$A_{\max}(t) = B_{eq}(t) + \frac{B_f(t) + A_f(t) - B_{eq}(t)}{N_b(t) + 1} \quad (\text{만족 상태})$$

여기서 $A_{\max}(t)$ 는 접속에 할당해 줄 수 있는 최대 할당량, $B_{eq}(t)$ 는 ABR 서비스를 위한 할당된 전체 대역폭을 전체 접속의 수로 나눈 동등한 몫(Equal Share), $N_b(t)$ 는 불만족 상태의 접속의 수, $B_f(t)$ 는 ABR 서비스를 위한 전체 대역폭에서 불만족 상태의 접속들의 $B_{eq}(t)$ 의 합을 뺀, 즉 자유 대역폭(free bandwidth)을 말하며, $A_f(t)$ 는 현 접속에 할당된 대역폭을 의미한다. 이 알고리즘은 최대-최소 할당에 따라 전송률을 할당하지만 MCR에 비례한 할당 및 MCR을 보장하지 않는다는 단점이 있다. ATM Forum에서는 MCR과 관련된 여러 가지 대역폭 할당 방법을 제안하고 있다[1]. 제안 중에서 MCR에 비례한 할당에 대하여 알아보자. 이 알고리즘은 MCR에 비례하여 용량을 할당한다. 할당하는 알고리즘을 식으로 표현하면 아래와 같다.

$$A_{\max}(t) = B(t) * \left(\frac{MCR_i}{\sum_{i \in \text{불만족상태}} MCR_i} \right)$$

여기서 $B(t)$ 는 ABR 서비스를 위한 할당된 대역폭, MCR_i 는 접속의 MCR, $\sum_{i \in \text{불만족상태}} MCR_i$ 는 불만족 상태의 접속들의 MCR의 합을 의미한다. 즉 접속 i 에 대하여 ABR 서비스를 위한 대역폭에서 만족 상태인 접속의 대역폭을 MCR에 비례하여 할당하는 것이다. ATM 포럼에서 제시한 대역폭 할당은 접속들의 MCR에 비례하여 대역폭을 할당하지만 MCR을 보장하지 않는다는 단점이 있다.

본 논문에서는 MCR을 보장하고 큐 제어 함수를 사용하여 큐 길이를 일정하게 유지함으로써 일정한 셀 지연시간을 가지며 접속의 상태를 만족과 불만족 상태로 나눔으로써 대역폭 활용도를 높일 수 있는 알고리즘을 제안한다.

2. 대역폭 할당 알고리즘

VOD 서비스처럼 가변적인 전송률을 가지면서 일정한 지연시간을 요구하고 MCR이 보장되어야하는 서비스는 ABR을 많이 이용할 걸로 예상된다. 그러므로 본 논문에서는 MCR

을 보장하고 전송률을 MCR에 비례해서 할당하며 큐 함수를 사용해서 큐 길이 및 일정한 전송 지연을 유지하는 전송률 할당 알고리즘을 제안한다. 알고리즘의 설명 및 슈도 코드를 위해서 사용되는 용어는 다음과 같다

MCR^i	접속 i 의 MCR
B_T	ABR 서비스를 위한 대역폭
B_S	만족 상태인 접속에 할당된 대역폭
B_b^{MCR}	불만족 상태인 접속들의 MCR의 합
ACR_{cell}^i	RM셀의 ACR 항목, 접속에 할당된 대역폭
ER_{cell}^i	접속 i 에 속한 RM셀의 ER 항목
CCR_{sw}^i	스위치에서 측정된 접속 i 의 전송률
A_{max}	공정한 몫

[표 1] 알고리즘에 사용되는 용어

2.1 공정한 몫

현재 MIT기법[4]을 기반으로 한 여러 전송률 할당 알고리즘은 접속들을 만족 상태와 불만족 상태로 구분하고 ABR을 위한 대역폭에서 만족 상태인 접속들에 할당된 총합을 뺀 나머지를 불만족 상태에 있는 접속들의 개수로 나누어서 공정한 몫을 구하며, 공정한 몫보다 큰 대역폭을 요구할 경우 공정한 몫으로 전송률을 할당하게 된다. 이렇게 될 경우 공정한 몫보다 큰 대역폭을 요구하는 접속은 대역폭의 요구량과 관계없이 공정한 몫으로 전송률을 할당하므로 정보 전송량에 따라 과금을 수행하는 미래의 광대역 통신망에서는 단점으로 여겨질 수 있다. 그러므로 본 논문에서는 아래와 같이 각 접속들의 MCR값에 비례하여 전송량을 할당한다.

IF(connection i == 불만족 상태)

$$A_{max} = (B_T - B_S) \times \frac{MCR^i}{B_b^{MCR}}$$

ELSE / connection i == 만족 상태 */*

$$A_{max} = (B_T - B_S + ACR_{sw}^i) \times \frac{MCR^i}{B_b^{MCR} + MCR^i}$$

위의 식은 MCR을 고려하지 않은 식이다. MCR을 고려한 전송률 할당기준은 ATM 포럼에서 제시한 공정성 기준의 예에서 두 가지[1]가 해당되며 다음과 같다.

2.1.1 MCR과 동등한 몫의 합

접속에 할당되는 대역폭은 모든 접속의 MCR에 할당된 대역폭을 제외한 나머지 대역폭을 동등하게 나눈 몫과 MCR의 합이다.

$$Allocated_i = MCR_i + \frac{B_T - \sum_j MCR_j}{N}$$

2.1.2 MCR과 최대-최소 공정한 몫(max-min fairshare)간의 최대값

접속에 할당되는 대역폭은 MCR과 최대-최소 공정한 몫 중에서 더 큰 값이 된다.

$$Allocated_i = \max(MCR_i, \max_min \text{ fairshare})$$

MCR과 최대-최소 공정한 몫(max-min fairshare)간의 최대값을 이용한 할당은 MCR은 보장하지만 여전히 공정한 몫보다 큰 대역폭을 요구할 경우 공정한 몫으로 할당하게 된다. 또한 MCR과 동등한 몫의(equal share)의 합을 이용한 할당도 마찬가지로 MCR은 보장하지만 MCR을 제외한 나머지 대역폭을 모든 접속에 똑같이 할당하게 된다. 본 논문에서 제시하는 알고리즘은 MCR을 보장하면서 접속의 MCR에 비례한 전송률을 할당하기 위한 공정한 몫을 계산식은 다음과 같다.

IF(connection i == 불만족 상태)

$$A_{\max} = MCR^i + (B_T - B_b^{MCR} - B_S) \times \frac{MCR^i}{B_b^{MCR}}$$

ELSE /* connection i == 만족 상태 */

$$A_{\max} = MCR^i + (B_T - B_b^{MCR} + B_S + ACR_{sw}^i - MCR^i) \times \frac{MCR^i}{B_b^{MCR} + MCR^i}$$

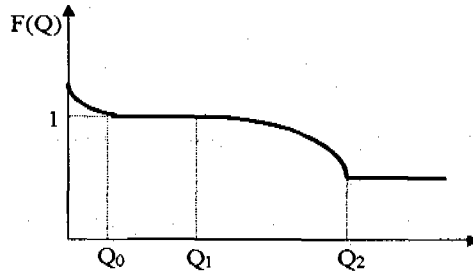
위의 식은 ABR을 위한 대역폭에서 모든 접속의 MCR을 제외한 나머지 대역폭에 MCR에 비례한 값을 곱하고 접속의 MCR을 더하게 된다. 이렇게 함으로서 MCR을 보장 할뿐만 아니라 나머지 대역폭을 MCR에 비례하여 할당하게 된다.

2.2 큐 제어 함수

제안하는 알고리즘은 ABR을 위한 대역폭의 85 ~ 95%만을 접속의 대역폭으로 할당함으로써 대역폭 활용도를 높이고 혼잡을 방지한다. 다음은 큐 제어 함수를 식으로 나타낸 것이다.

$$F(Q) \begin{cases} = \frac{\lambda_b U_0}{(\lambda_b - 1)Q + Q_0} & 0 \leq Q < Q_0 \quad (1) \\ = 1 & Q_0 \leq Q < Q_1 \quad (2) \\ = 2 - \frac{\lambda_a Q_1}{(\lambda_a - 1)Q_1 - Q} & Q_1 \leq Q < Q_2 \quad (3) \\ = 2 - \frac{\lambda_a Q_1}{(\lambda_a - 1)Q_1 - Q_2} & Q_2 \leq Q < \infty \quad (4) \end{cases}$$

큐 제어 함수는 큐 길이에 따라서 총 4 단계로 나뉘어 지며 다음 [그림 1]은 큐 제어 함수의 개념을 보여주고 있다.



[그림 1] 큐 제어함수의 그래프

Q가 Q0보다 작을 때 식(1)이 적용되며 F(Q)는 1보다 큰 값이 되어 전송률을 증가 시켜서 Q1과 Q2사이에 들게 한다. Q가 Q0와 Q1사이일 때는 식(2)가 적용되며 F(Q)는 1이 되어 큐 제어 함수는 전송률 할당에 어떠한 영향도 미치지 못하게 되며 Q가 Q1과 Q2사이일 때는 식(3)이 적용되며 F(Q)는 1보다 작은 값이 되어 전송률을 감소 시켜서 Q1과 Q2사이에 들게 한다. Q가 Q2보다 클 때는 식(4)가 적용되며 F(Q)는 1보다 매우 작은 값이 되어 전송률을 빠르게 감소시켜서 큐 길이가 빠르게 Q1과 Q2사이에 들게 한다. 또한 Q0보다 작을 때 와 Q1과 Q2사이일 때 Q가 Q0와 Q1에 가까워 질 수록 F(Q)는 1에 가까워져서 Q가 Q0와 Q1사이에 들지 못하고 하여 불필요한 진동을 막는다.

2.3 순방향 알고리즘

$$\begin{aligned}
 & B_T = \text{link rate} - \text{VBR rate} - \text{CBR rate} \\
 & \text{if}(CCR_{sw}^i < ACR_{sw}^i) \\
 & \quad B_T = B_T + ACR_{sw}^i - CCR_{sw}^i \\
 & \text{if}(\text{connection } i = \text{만족 상태}) \\
 & \{ \\
 & \quad \text{connection } i = \text{불만족 상태} \\
 & \quad B_S = B_S - ACR_{sw}^i \\
 & \quad B_b^{MCR} = B_b^{MCR} + MCR^i \\
 & \}
 \end{aligned}$$

[그림 2] 순방향 알고리즘

링크의 용량에서 CBR과 VBR의 총 전송률을 뺀 나머지 대역폭을 ABR을 위한 대역폭(BT, 목적 전송률)으로 할당하게 된다. 만약 소스가 할당된 전송률 보다 작게 전송하고 있다면 대역폭 활용도가 낮아질 수 있으므로 할당된 전송률과 실제 전송률과의 차이를 다른 접속이 사용할 수 있도록 하기 위해 현재의 전송률을 측정하고 할당된 전송률보다 작은지를 검사해 만약 작다면 목적전송률을 할당된 전송률과 실제 전송률과의 차이만큼을 증가

시킨다. 그 다음 접속의 현재 상태가 만족 상태이면 상태를 불만족 상태로 바꾸고 스위치에 저장되는 만족 상태인 접속들에 할당된 대역폭 B_S 와 불만족 상태인 접속들의 MCR의 합 (B_b^{MCR})을 새롭게 갱신한다. 이러한 과정들이 모두 끝난 후 RM셀의 내용 중 어떠한 것도 변경시키지 않고 다음 스위치로 전송한다. 즉, 순방향 RM셀이 스위치에 도착했을 때 하는 일은 목적전송률을 구하고 접속의 현재 전송량 (CCR_{sw}^i)이 접속에 할당된 대역폭보다 작을 경우에 목적전송률을 접속에 할당된 대역폭 (ACR_{cell}^i)과 현재 전송량의 차이만큼 증가시킨다.

2.4 역방향 알고리즘

```

if(connection i == 불만족 상태)
{

$$A_{\max} = MCR^i + (B_T - B_b^{MCR} - B_s) \times \frac{MCR^i}{B_b^{MCR}}$$

if( $A_{\max} > ACR_{cell}^i$ )
{
connection i = 만족 상태

$$B_S = B_S + ACR_{cell}^i$$


$$B_b^{MCR} = B_b^{MCR} - MCR^i$$

}
}
else /* connection i == 만족 상태 */
{

$$A_{\max} = MCR^i + (B_T - B_b^{MCR} - B_s + ACR_{sw}^i - MCR^i) \times \frac{MCR^i}{B_b^{MCR} + MCR^i}$$

if( $A_{\max} < ACR_{cell}^i$ )
{
connection i == 불만족 상태

$$B_S = B_S - ACR_{sw}^i$$


$$B_b^{MCR} = B_b^{MCR} + MCR^i$$

}
else

$$B_S = B_S - ACR_{sw}^i + ACR_{cell}^i$$

}

$$ACR_{sw}^i = ACR_{cell}^i$$


$$ER_{cell}^i = \min(ER_{cell}^i, \max(A_{\max} \times Queue\_function(Queue\ length), MCR))$$

```

[그림 3] 역방향 알고리즘

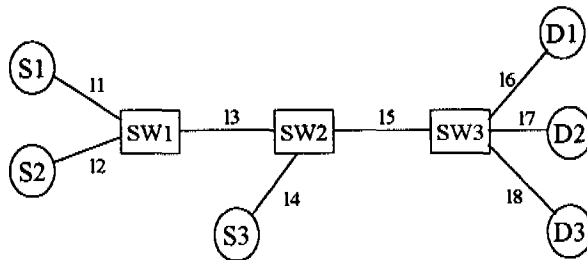
실제적인 전송률 할당은 역방향 RM셀이 스위치에 도착했을 경우에 일어난다. 접속의 상태가 만족 상태인지 불만족 상태인지를 판단해 상태에 따라 다른 알고리즘을 수행하게 된다. 접속 i 의 상태가 만족 상태라면 공정한 몫을 만족 상태일 경우의 공정한 몫 계산식에 따라 계산하고 계산된 공정한 몫이 접속에 할당된 대역폭과 비교해서 공정한 몫이 더 작다면 접속 i 의 상태를 불만족 상태로 바꾸고 만족 상태인 접속에 할당된 대역폭의 총합(BS)을 접속 i 에 할당된 대역폭만큼 감소시키고 불만족 상태인 접속들의 MCR의 합을 접속 i 의 MCR만큼 증가시킨다. 마찬가지로 접속 i 의 상태가 불만족 상태라면 공정한 몫을 불만족 상태일 경우의 공정한 몫 계산식에 따라 계산하고 공정한 몫이 접속에 할당된 대역폭과 비교해서 공정한 몫이 더 크다면 접속 i 의 상태를 만족 상태로 바꾸고 만족 상태인 접속에 할당된 대역폭의 총합과 불만족 상태인 접속들의 MCR의 합을 비슷한 방법으로 갱신하고 RM셀의 ACR항목을 다음 RM셀이 도착했을 때 변수들의 갱신을 위해 스위치에 저장한다. 위의 모든 과정이 끝난 후에 실제적인 전송률이 할당되는데 계산된 공정한 몫에 큐 제어 함수의 리턴 값을 곱하고 MCR값과 비교(큐 제어 함수에 의해 MCR값보다 더 작아 질 수 있음)해 더 큰 값을 셀에 있는 ER값과 비교해 더 작은 값을 전송률로 할당하게 된다. 그리고 나서 RM셀을 다음 스위치로 전송하게 된다.

3. 성능 평가

본 논문에서 제안한 알고리즘의 성능을 ATM포럼에서 제시한 MCR에 비례한 할당과 Kalampoukas 논문과 비교해서 평가한다. 제안하는 알고리즘의 성능평가는 MCR의 보장, 소스가 할당된 전송량보다 적게 전송할 경우에 대역활용도의 차이, 접속의 상태를 만족 상태와 불만족 상태로 나눔으로써 생기는 대역활용도의 차이, 큐 제어 함수의 사용에 의한 큐 길이의 일정성 등에 대한 네 가지의 항목에 대해서 이루어진다.

3.1 시뮬레이션 환경

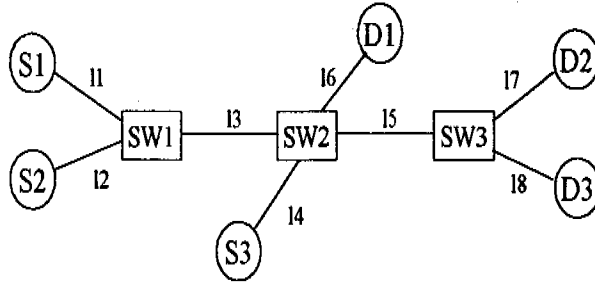
성능평가를 위한 시뮬레이션 환경은 평가하는 항목에 따라 두 가지 환경을 사용한다. 먼 첫 번째 환경은 [그림 4]와 같다.



[그림 4] 시뮬레이션 환경1

위 환경은 MCR의 보장, 접속이 할당된 전송량보다 적게 전송할 경우의 대역활용도, 큐 합

수의 영향등 세가지 항목을 평가하기 위해 사용되며 세 개의 소스와 세 개의 데스티네이션 그리고 세 개의 스위치로 구성된다. 각각은 서로 적절하게 155 Mbps의 링크로 서로 연결되며 이때 링크 15가 가장 혼잡이 심한 링크가 된다. 두 번째의 시뮬레이션 환경은 [그림 5]이다.

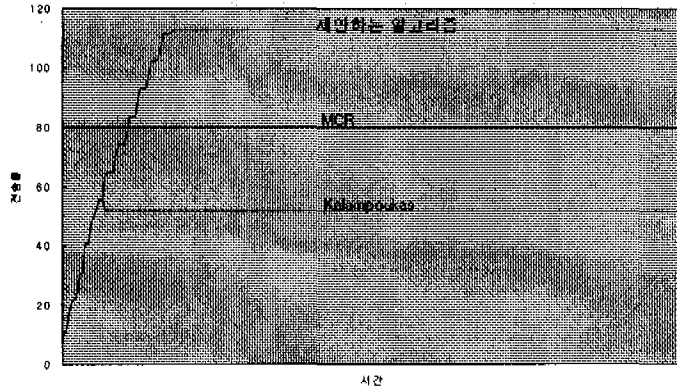


[그림 5] 시뮬레이션 환경2

위 환경은 접속을 만족 상태와 불만족 상태로 나눔으로써 생길 대역활용도의 개선에 대한 평가를 위해서 사용되며 [그림 4]와 마찬가지로 세 개의 소스와 세 개의 데스티네이션 그리고 세 개의 스위치로 구성된다. 각각은 서로 적절하게 155 Mbps의 링크로 서로 연결된다. 그러나 [그림 4]와 다른 점은 S1이 15를 거치지 않고 SW2에서 바로 D1로 연결되었다.

3.2 MCR의 보장

VOD나 AOD와 같은 서비스는 일정한 셀 지연도 중요하지만 MCR의 보장도 매우 중요하게 여겨진다. Kalampoukas 알고리즘은 MCR을 고려하지 않았기 때문에 MCR을 보장하지 못하지만, 본 논문에서 제안하는 알고리즘은 MCR을 보장한다. MCR의 보장에 대한 평가를 위해 시뮬레이션 환경 1을 사용하며 각 소스의 MCR을 S1은 10 Mbps, S2는 20Mbps, S3은 80Mbps로 하고 S1의 실제 전송률인 14의 전송률을 Kalampoukas와 본 논문에서 제안하는 알고리즘과 비교한다.

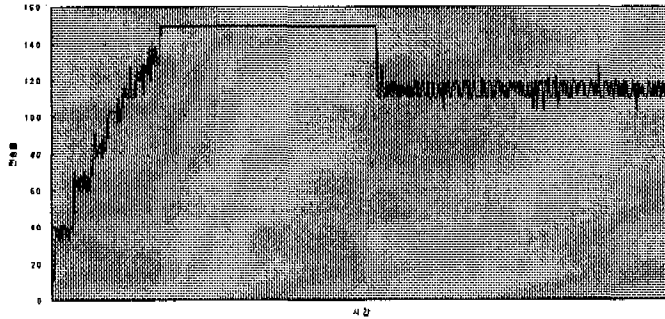


[그림 6] MCR의 보장에 대한 평가

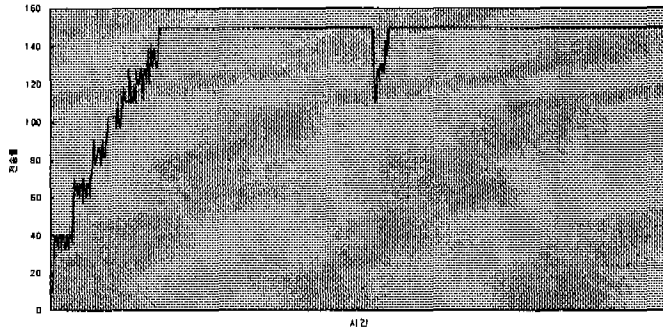
[그림 6]은 망이 안정상태에 도달했을 때 Kalampoukas는 모든 접속에 약 50Mbps를 동일하게 할당하기 때문에 MCR을 보장하지 못하고 있다. 반면 제안하는 알고리즘은 MCR을 보장하고 있음을 보여주고 있다.

3.3 할당된 전송량보다 작게 전송할 경우 대역활용도의 평가

소스가 망으로부터 할당된 전송량보다 작게 전송할 경우 대역활용도가 낮아진다. 그러므로 할당된 전송량보다 작게 전송할 경우 그만큼의 차이를 다른 대역폭이 사용할 수 있다면 대역 활용도가 높아질 것이다.



[그림 7] ATM 포럼 MCR에 비례한 할당 (15의 전송률)



[그림 8] 제안하는 알고리즘 (15의 전송률)

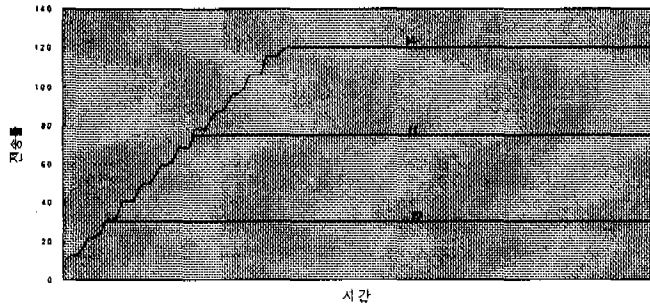
본 항목에 대한 평가는 시뮬레이션 환경 1을 사용하고 소스의 MCR을 각각 S1은 10 Mbps, S2는 20 Mbps, S3은 40 Mbps로 주고 S3의 전송률을 어느 순간 50 Mbps로 감소시킬 경우에 가장 병목이 심한 15에서의 전송률을 검사해서 ATM 포럼 MCR에 비례한 할당과 제안하는 알고리즘의 대역활용도를 비교 평가한다. [그림 6]과 [그림 8]은 본 항목에 대한 평가 결과를 보여주고 있다. MCR을 각각 10, 20, 40 Mbps로 하였으므로 ATM 포럼 MCR에 비례한 할당과 제안하는 알고리즘은 S1, S2, S3에 약 21, 42, 84 Mbps를 할당한다. 그러나 S3의 전송률을 어느 순간 50Mbps로 감소시켰을 경우에 ATM 포럼 MCR에 비례한 할당은 할당된 전송률과의 차이 34Mbps를 다른 접속이 사용하지 못하므로 15의 대역폭의 낭비가 있음을 볼 수가 있다. 반면 제안하는 알고리즘은 S3의 50Mbps로 감소시켰을 때 할당된 전송률과의 차이 34 Mbps를 다른 접속이 사용할 수 있으므로 15의 전송률이 일시적으로 감소했다가 149 Mbps까지 증가하는 것을 볼 수 있다.

3.4. 만족, 불만족 상태에 의한 대역폭 활용도의 개선

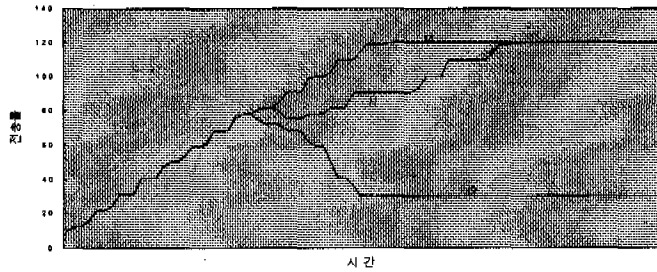
ATM 포럼의 MCR에 비례한 할당은 다른 스위치에서의 혼잡으로 인해 공정한 몫보다 작게 할당될 수가 있으므로 대역폭의 낭비가 생긴다. 예를 들어 시뮬레이션 환경 2에서 S1, S2, S3의 MCR이 각각 10, 20, 80 Mbps로 할 때 SW1에서는 S1에 75, S2에 75 Mbps를 할당하고 SW2에서는 S2에 37.5, S3에 112.5 Mbps를 할당한다. 소스에 할당되는 대역폭은 병목이 가장 심한 스위치에서 할당된 대역폭이 되므로 결국 S1에는 75, S2에는 37.5, S3에는 112.5 Mbps가 할당된다. S1과 S2는 13을 통과하게 되고 13 링크용량의 37.5 Mbps가 낭비되게 된다. 직관적으로 보면 S1은 15를 통과하지 않으므로 낭비되는 37.5 Mbps를 S1에 할당할 수 있음을 알 수 있다. 제안하는 알고리즘은 공정한 몫보다 작게 할당된 접속의 상태를 만족 상태로 공정한 몫으로 할당된 접속을 불만족 상태로 구분함으로써 낭비되는 37.5 Mbps를 S1에 할당하게 되고 대역폭 활용도를 높인다. 이를 평가하기 위해서 시뮬레이션 환경 2에서는 S1, S2, S3의 MCR이 각각 10, 20, 80 Mbps로 하여 각 소스의 전송률 11, 12, 14과 ATM 포럼 할당방법에서 대역폭 낭비가 일어나는 13의 전송률을 제안한 알고리즘과 비교를 하였다.

[그림 9]와 [그림 10]에서 보듯이 ATM포럼 MCR에 비례한 할당[그림 9]는 S1의 전송률이 11의 전송률은 안정상태에서 75Mbps이지만 제안하는 알고리즘[그림 11]은 112.5Mbps가

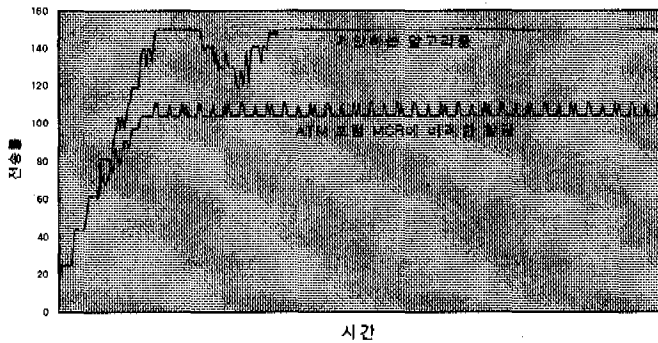
됨을 알 수 있다. [그림 11]은 ATM 포럼 MCR에 비례한 할당에서 대역폭의 낭비가 발생하는 13의 전송률을 제안하는 알고리즘과 비교한 것이다. 그림에서 보듯이 13의 전송률은 149 Mbps까지 전송할 수 있음에도 불구하고 ATM 포럼 MCR에 비례한 할당은 약 37.5 Mbps의 낭비가 생김을 알 수 있다. 반면 제안하는 알고리즘은 일시적으로 대역폭의 낭비가 생기지만 안정 상태에서는 13 대역폭의 모두를 사용하고 있음을 보여주고 있다. 이는 접속의 상태를 만족 상태와 불만족 상태로 나누어서 낭비되는 37.5 Mbps의 대역폭을 S1에 할당했기 때문이다.



[그림 9] ATM 포럼 MCR에 비례한 할당 (11, 12, 14의 전송률)



[그림 10] 제안하는 알고리즘 (11, 12, 14의 전송률)

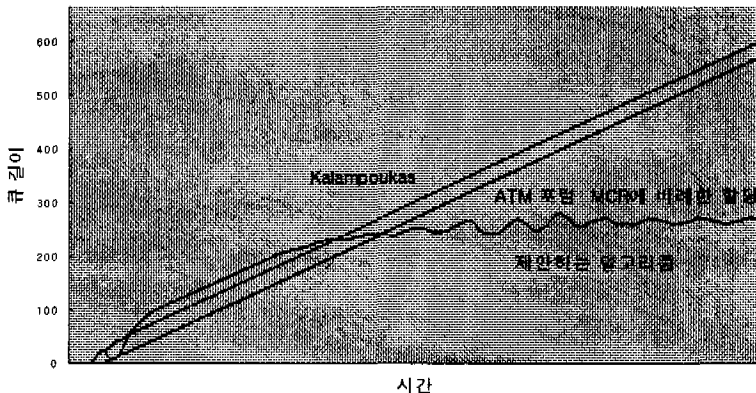


[그림 11] 13의 전송률 비교

3.5 큐 제어 함수의 영향평가

서비스 품질의 저하를 막기 위해서 되도록 작은 셀 지연변이를 유지하는 것이 중요하다. 그러므로 본 논문에서는 큐 제어 함수를 사용해서 큐의 길이를 일정하게 유지하도록 하고 또한 셀 지연을 일정하게 유지한다. 큐 제어 함수에 의한 큐 길이의 영향을 평가하기 위해 시뮬레이션 환경 1을 사용하며 목적전송률을 155 Mbps로 과다할당이 일어나도록 했을 때의 큐 길이를 Kalampoukas, ATM 포럼 MCR에 비례한 할당, 제안하는 알고리즘과 비교한다. 큐 길이는 시뮬레이션 환경 1에서 가장 병목이 심한 SW2에서 측정하고 제어함수의 파라미터들은 다음과 같다.

Parameter : $\lambda a = 2$, $\lambda b = 1.3$, $Q_0 = 100$, $Q_1 = 200$, $Q_2 = 300$



[그림 12] 큐 길이의 비교

[그림 12]에서 보듯이 Kalampoukas와 ATM 포럼 MCR에 비례한 할당은 큐 길이가 일정하게 계속 증가되는 것을 볼 수 있다. 반면 제안하는 알고리즘은 100까지는 두 알고리즘보다 빠르게 증가하다가 100과 200사이에서는 큐 제어 함수가 전송률 할당에 아무런 영향을 미치지 못하기 때문에 일정하게 증가한다. 200이 넘었을 경우에는 큐 길이가 더 이상 증가하지 못하고 거의 일정하게 유지되는 것을 볼 수 있다.

4. 결론

본 논문에서는 ABR을 이용해 VOD와 같은 최소 셀 전송률(MCR)과 일정한 전송지연을 요하는 서비스를 지원할 수 있는 전송률 할당 알고리즘을 제안하였다. 물론 CBR이나 VBR을 이용해 서비스할 수도 있지만 대역폭 활용도가 매우 낮다는 단점[8, 9] 때문에 ABR을 이용하여 서비스하려는 연구가 최근 활발히 진행되고 있고 본 논문도 이러한 범주에 포함된다. 제안하는 알고리즘은 MCR을 보장할 뿐만 아니라 공정한 몫의 개념을 달리 해 MCR에 비례하여 전송률을 할당한다. 접속의 상태를 만족 상태와 불만족 상태로 구분

함으로써 만족 상태의 접속이 사용하지 않는 나머지 대역폭을 불만족 상태인 접속이 사용할 수 있도록 해 대역폭 활용도를 높인다. 또한 접속에 할당된 전송률보다 작게 전송할 경우 할당된 전송률과의 차이를 다른 접속이 사용할 수 있도록 해 대역폭 활용도를 높였다. 그리고 큐 제어 함수의 사용으로 인해 항상 일정한 큐 길이를 유지할 수 있을 뿐만 아니라 일정한 셀 지연을 유지할 수 있다. 본 논문에서 제안하는 알고리즘을 수정된 NIST ATM simulator[11]를 이용해 ATM 포럼에서 제시한 MCR에 비례한 할당과, Kalampoukas 기법과 위에서 제시한 네 가지 항목 즉 MCR에 비례한 할당, 접속에 할당된 전송률보다 작게 전송할 경우, 접속의 상태를 만족 상태와 불만족 상태로 구분, 큐 제어 함수에 의한 큐 길이 등에 관해서 시뮬레이션을 통해 제안한 알고리즘의 타당함을 증명하였다.

소스가 RM셀을 보낼 때 PCR이 아닌 소스가 다음에 보낼 정보의 전송량으로 ER값을 셋팅 한다면 공정한 룩의 개념을 MCR에 비례한 할당이 아닌 실제 소스가 전송하길 원하는 전송량에 비례하여 전송량을 할당할 수 있을 것이다. 그러므로 앞으로 해야 할 일은 소스 측의 전송량을 미리 예측할 수 있는 알고리즘을 개발하는 것이다.

[참 고 문 헌]

- [1] ATM Forum : Traffic Management Specification Version 4.0 af-tm-0056.000, April 1996
- [2] L. Kalampoukas, "An Efficient Rate Allocation Algorithm for ATM Networks", UCSC MS Thesis, May 1995
- [3] R. Jain, S. Kalyanaraman, S. Fahmy, "Source Behavior for ATM ABR Traffic Management: An Explanation", IEEE Communications Magazine, November 1996
- [4] A. Charny, "Congestion Control With Explicit Rate Indication", ICC'95, June 1995
- [5] N. Ghani, W. Mark, "Queueing Analysis of a Distributed Explicit Rate Allocation Algorithm for ABR Service", IEEE INFOCOM'97, 1997
- [6] R. Jain, S. Kalyanaraman, "The OSU Scheme for Congestion Avoidance in ATM networks Using Explicit rate Indication", Proceedings WATM'95 First Workshop on ATM Traffic Management, Paris, December 1995
- [7] L. Roberts, "Enhanced PRCA(Proportional Rate-Control Algorithm)", ATM Forum 94-075, August 1994
- [8] A. W. Barnhart, "Explicit Rate Performance Evaluation", ATM Forum Traffic Management 94-00983R1, Sep 1994
- [9] H. Kanakia, P. P. Mishra, A. Reibman, "An Adaptive Congestion Control Scheme for Real Time Packet Video Transport", SIGCOMM'93, 1993
- [10] M. Hiuchvj, "Closed-Loop Rate-Based Traffic Management", ATM Forum Traffic Management 94-0211R3, Apr 1994
- [11] N. Golmie, A. Koenig, D. Su, "The NIST ATM Network Simulator", Technical Report NISTIIR 5703, March 1998