

# PGMCC의 공정성 향상

박 영 선<sup>†</sup> · 현 도 원<sup>\*\*</sup> · 장 주 옥<sup>\*\*\*</sup>

## 요 약

본 논문은 멀티캐스트 혼잡제어방식 제안 중의 하나인 PGMCC의 공정성 향상 방법에 관한 것이다. 안정적인 멀티캐스트 전송과 혼잡제어를 위한 여러 가지 방법중에서 PGMCC는 유력한 방식으로 제안되었다. 하지만 PGMCC에도 고정된 타임 아웃 값의 사용과 불확실한 Acker 선출 등 공정성을 낮추는 요인들이 몇 가지 지적되고 있다. 본 논문에서는 타임 아웃 값을 네트워크 상황에 맞추어 적응적으로 가변시키고, 라우터에서의 쓰루풋 비교를 통한 선택적인 NAK 억압이라는 Acker 선출 방법을 도입하여 이와 같은 문제를 해결하여 공정성이 향상되는 것을 보인다.

## Enhancing the Fairness of PGMCC

Young-sun Park<sup>†</sup> · Do-won Hyun<sup>\*\*</sup> · Ju-wook Jang<sup>\*\*\*</sup>

### ABSTRACT

To deploy multicast protocols, fairness to current Internet traffic, particularly TCP, is an important requirement. PGMCC is one of the most promising multicast congestion control proposals but it suffers from degradation of fairness by fixed timeout and uncertain acker selection. In this paper, we suggest addition of an adaptive timeout mechanism and NAK suppression in router using throughput comparison to improve fairness. Our simulation show improved fairness.

키워드 : 멀티캐스트(Multicast), PGMCC, 혼잡제어(Congestion Control), 공정성(Fairness)

### 1. 서 론

멀티캐스트는 다수의 사용자에게 동일한 데이터를 전송할 때 하나의 링크에 동일한 데이터 스트림이 두 번이상 지나 가지 않게 함으로서 대역폭을 절감하는 전송 방법으로서 다수에게 멀티미디어 전송을하는 VOD(Video-On-Demand), 실시간 인터넷 원격강의 시스템과 같은 경우에 데이터의량을 획기적으로 줄일 수 있게 한다[1]. 그러나 기존에 널리 사용되고 있는 TCP(Transmission Control Protocol) 트래픽의 경우에는 혼잡제어가 되는 반면 멀티캐스트 트래픽의 경우에는 혼잡제어가 이루어지지 않기 때문에 같은 인터넷 망에서 공존할 수가 없다. 이러한 문제점을 해결하기 위하여 RMTP(Reliable Multicast Transport Protocol), TFMCC(TCP-Friendly Multicast Congestion Control), PGM(Pragmatic General Multicast)등의 여러 가지 멀티캐스트 방식들이 제안되고 있다.

이런 방식중에서 PGMCC(Pragmatic General Multicast Congestion Control)[2]는 단일 속도 제어의 대표자 기반 멀티캐스트 혼잡제어 방식이다. PGMCC는 Acker라 불리는

대표자와의 데이터 교환을 통해 전송 윈도우의 크기를 조절한다. Acker는 수신자중에서 가장 낮은 전송율을 가지는 수신자가 Acker로 선출된다. PGMCC는 현재 NAK 중심의 멀티캐스트 혼잡제어 방식중에서 유망한 제안중의 하나지만 몇 가지의 단점을 가지고 있다. PGMCC의 경우 타임 아웃(timeout) 값이 일정하기 때문에 선택된 Acker의 RTT(Round-Trip Time)에 따라서 공정성이 심각하게 손상될 수 있다. 또한 Acker의 선택에 있어서 NAK 억압에 의해서 올바른 Acker가 선택되지 않을 수 있다.

본 논문에서는 이러한 단점을 보완하기 위해서 TCP와 같은 적응적인 타임 아웃(adaptive timeout) 방식을 적용하였고, Acker 선택 방법을 추가 보완하였다. 다음 2절에서는 멀티캐스트 혼잡제어와 PGMCC의 문제점에 대하여 설명하고 3절에서는 본 논문에서 제안된 새로운 방법, 4절에서는 기존의 방법과 제안된 새로운 방법을 통하여 얻은 결과를 분석하였고, 5절에서는 결과와 추후 연구방향에 대해서 서술하였다.

### 2. 기존 PGMCC의 문제점

#### 2.1 PGMCC

PGM[3]은 단일 송신자 방식의 멀티캐스트 방식으로 NAK를 기반으로한 재전송 요청을 통하여 안정적인 멀티캐스트

<sup>†</sup> 준 회원 : LG전자 Digital & Multimedia 연구소 연구원

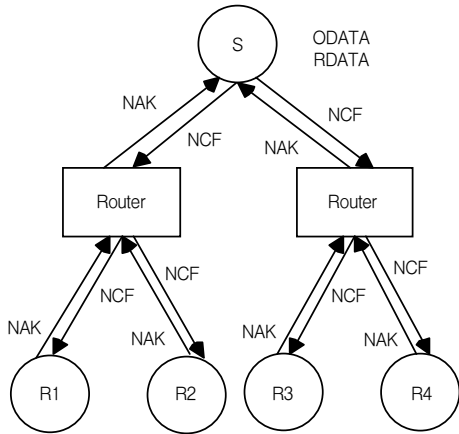
<sup>\*\*</sup> 준 회원 : 서강대학교 대학원 전자공학과

<sup>\*\*\*</sup> 정 회원 : 서강대학교 전자공학과 교수

논문접수 : 2003년 1월 10일, 심사완료 : 2003년 3월 25일

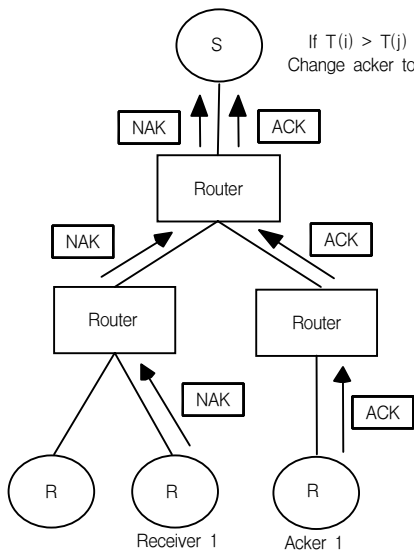
를 제공한다. PGM에서는 제한 역압을 통하여 멀티캐스트 망의 확장성을 제공한다.

(그림 1)은 PGM의 시나리오를 보여주고 있다. 송신자는 ODATA(Original Data)를 각 수신자에게 멀티캐스트로 전송한다. 이때, 데이터를 받지 못한 수신자는 송신자를 향해서 NAK를 전송하게 된다. 이 때 라우터는 먼저 받은 NAK를 위의 라우터로 전송하고 밑의 수신단에는 더이상 NAK를 전송하지 말라는 NCF(NAK Confirmation) 메시지를 전달하게 된다. 만약에 같은 NAK를 이 라우터가 받았을 경우에는 NAK를 전송하지 않고 역압하게 된다.



(그림 1) PGM 시나리오

PGMCC는 PGM에서 혼잡제어를 하기 위하여 제안된 방식이다. Acker라고 하는 대표자를 선택하여 송신자와 이 대표자 사이에는 제어 관계를 유지한다. 이러한 대표자를 Acker라고 부르는 이유는 대표자만이 ACK를 보내기 때문이다. 다른 수신자의 경우 패킷이 손실되었을 경우에만 NAK를 보낼 수 있다.



(그림 2) PGMCC 시나리오

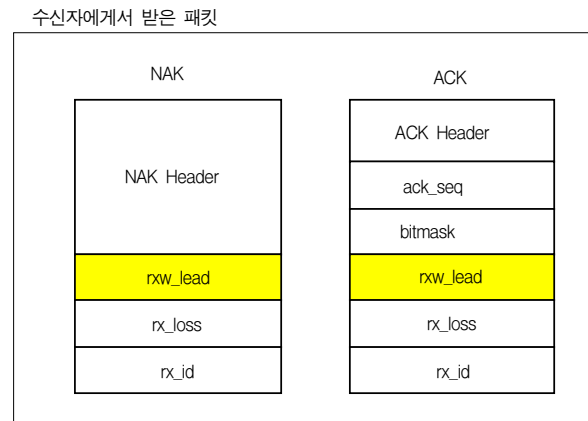
간단한 시나리오를 (그림 2)에서 확인 할 수 있다. Acker는 그룹의 대표자로서 가장 나쁜 쓰루풋(throughput)을 가지는 수신자가 선택된다. 송신자와 Acker 사이의 전송률은 TCP와 비슷한 윈도우(window) 기반으로 결정되며, 이는 역시 ACK에 의하여 결정된다. PGMCC에서 수신자가 쓰루풋을 판단하는 것은 수신자의 보고에 의해서 결정된다. 수신자는 NAK나 ACK에 들어있는 손실률(loss rate)과 시퀀스 번호(sequence number)를 이용하여 RTT를 측정 쓰루풋을 결정하게 된다. 손실률은 진동을 피하기 위하여 저대역필터를 사용하여 계산을 한다.

PGMCC에서 가장 중요한 점은 Acker의 선택에 있다. Acker는 가장 낮은 쓰루풋을 가지는 수신자가 선택된다. Acker의 선택은 쓰루풋의 비교를 통해서 결정한다.

$$T = C \frac{1}{RTT \sqrt{p}} \tag{1}$$

- T : throughput
- RTT : round trip time
- C : 상수 PGMCC의 경우 0.9[2]
- p : Loss rate

이를 계산하여 기존의 대표자 *i*와 수신자 *j*로부터 받은 NAK을 기반으로 하여  $T(i) > T(j)$ 일 경우 수신자 *i*에서 수신자 *j*로 대표자가 바뀌게 된다. (그림 3)은 RTT측정에 대한 그림이다.



rx\_lead = 수신자가 가장 최근에 받은 패킷의 시퀀스 번호  
 각 수신자의 RTT = 송신자가 가장 최근에 보낸 패킷의 시퀀스 번호 - rx\_lead

(그림 3) 각 수신자의 RTT 측정

RTT 측정을 편하게 하기 위해서 PGMCC에서는 시퀀스 번호를 사용해서 RTT를 측정하게 된다. PGMCC에서의 RTT는 단순히 쓰루풋 비교를 위해서만 필요하기 때문에 각 수신자의 RTT를 타임스탬프(timestamp)를 사용하여 정확하게 측정하는 것이 아니라 시퀀스 번호의 차이로서 결정한다. 이를 위해서 PGMCC에서는 ACK나 NAK에 수신자가 가장 최근에 받은 패킷의 시퀀스 번호를 표기하고 수

신자에게서 온 패킷을 받았을 때, 가장 최근에 보낸 패킷과의 차를 RTT로 사용한다.

## 2.2 문제점

### 2.2.1 고정적인 타임 아웃 값

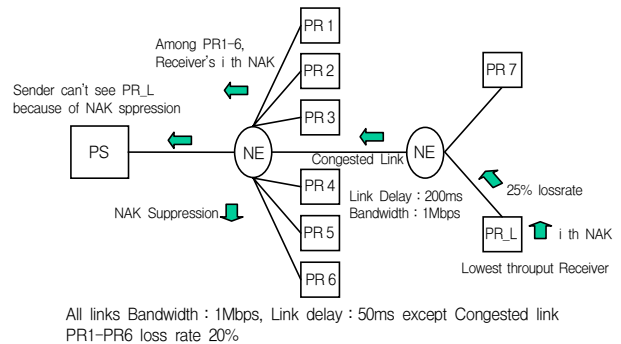
PGMCC는 타임 아웃값을 정하는 데에 있어 TCP와는 다르게 특별한 방법을 두지 않고 처음에 정한 값으로 계속 타임 아웃값을 유지하게 된다. PGMCC에서 고정적인 타임 아웃값을 사용하는 이유는 PGMCC의 경우 RTT의 측정을 위하여 타임스탬프를 사용하는 것이 아니라 시퀀스 번호를 사용하기 때문이다. PGMCC에서 RTT는 단지 대표자를 선출할 때, 수신자들간의 쓰루풋 비교를 위해 사용되기 때문에 시퀀스 번호를 사용해서 RTT를 측정하는 것이 더 낫다고 제안하고 있다. 하지만 이런 방법은 타임 아웃에 필요한 정확한 RTT를 산출해 낼 수 없기 때문에, TCP와 같은 적응적인 타임 아웃 값을 산출해 낼 수 없다. TCP와 같은 윈도우 방식을 사용하는 PGMCC의 경우에 타임 아웃값을 어떻게 정하느냐에 따라 공정성에 큰 영향을 미치게 된다. 만약, Acker의 RTT가 타임 아웃 값보다 작게 정해지면은 PGMCC의 경우 타임 아웃이 잘 안일어나기 때문에 TCP보다 윈도우 사이즈가 1로 잘 줄어들지 않게 되기 때문에 TCP는 상대적으로 낮은 쓰루풋을 가지게 된다. 반대로, Acker의 RTT가 타임 아웃 값보다 크게 되면은 계속해서 타임 아웃이 발생하기 때문에 전송 윈도우의 사이즈가 1이 되어 데이터를 제대로 전송하지 못하게 된다. 실제 네트워크의 경우 네트워크의 혼잡도는 다양하게 변하기 때문에 이런 고정적인 타임 아웃 값은 공정성에 큰 영향을 미칠 수가 있다[4].

### 2.2.2 NAK 억압에 의한 불확실한 Acker 선택

적응적인 타임 아웃 값을 적용하여 공정성의 향상을 가져올 수 있지만 특정한 경우에는 NAK 억압에 의해서 공정성은 저하될 수 있다. PGMCC는 가장 낮은 쓰루풋을 가지는 수신자를 대표자로 선택함으로써 TCP와 친화적인 공정성을 보장할 수 있다. PGM에서 제안된 케환 억압 방식은 가장 먼저 라우터에 도착한 NAK를 송신자에게 올려 보내고 다음에 도착하는 NAK의 경우 송신자에게 전송하지 않는다. 이 때, 비록 RTT는 크지만 패킷 손실률이 비슷한 수신자가 있을 경우, 송신자에 가까운 수신자는 NAK를 쉽게 송신자에게 보낼 수 있는 반면, 멀리 있는 수신자는 NAK가 가까이 있는 수신자보다 늦게 도착하여, 송신자에게 NAK를 전송할 수가 없다. 이때 멀리 있는 수신자의 쓰루풋은 알 수가 없으므로, 대표자 선택에서 제외되게 된다. 잘못된 Acker를 선정할 경우 공정성의 저하가 올 수 있다.

(그림 4)는 NAK 억압에 의한 Acker의 잘못된 선택에 대해서 보여주고 있다. 원래 Acker로 선택되어야 하는 PR\_L

수신자의 NAK는 송신자에 가까운 6개의 수신자에 의해서 NAK 억압이 되어 제대로 선택이 되지 않는다.



(그림 4) NAK 억압에 의한 Acker의 잘못된 선택

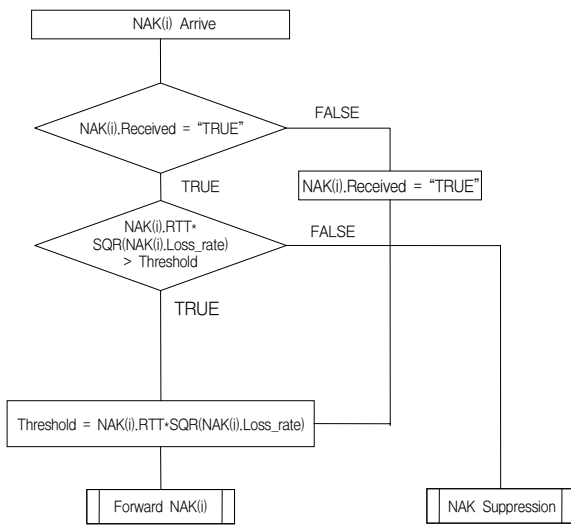
## 3. 제안된 방법

고정적인 타임 아웃 값에 의한 공정성 저하는 TCP와 같은 적응적인 타임 아웃 값을 적용함으로써 해결할 수 있다. 이 경우 기존의 PGMCC 논문에서 제안한 시퀀스 번호를 사용한 RTT 값이 아닌 타임 스탬프를 사용한 RTT 값을 사용하여야 한다. 시퀀스 번호를 사용한 RTT 값의 경우 쓰루풋 비교에 필요한 상대적인 RTT 값으로는 사용이 가능하지만 타임 아웃과 같이 절대적인 시간값을 구하는 데는 사용할 수 없기 때문이다. 식 (2)는 PGMCC에서 적용할려는 TCP의 타임 아웃 값 계산식이다.

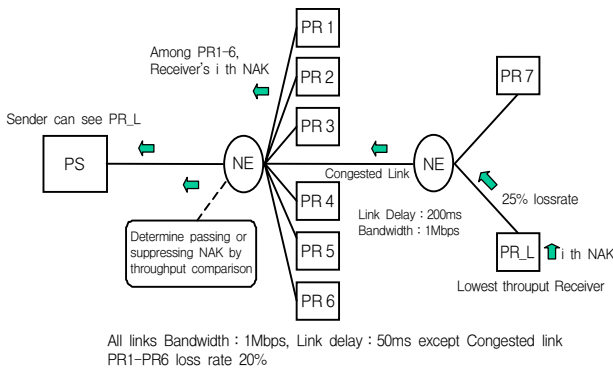
$$\begin{aligned}
 Err &= M - a \\
 A_{i+1} &= A_i + gErr \\
 D_{i+1} &= D_i + h(|Err| - D_i) \\
 \therefore RTO_{i+1} &= A_{i+1} + 4D_{i+1}
 \end{aligned}
 \tag{2}$$

A는 측정된 RTT, D는 RTT의 평균편차, Err은 현재 측정된 RTT와 전에 구했던 RTT와의 차이이다. 이득 g는 평균을 위한 것으로 0.125로 정해져 있다. 편차를 위한 이득 h는 0.25로 정해진다[5]. 타임 아웃 값이 고정적으로 정해져 있는 PGMCC에 적응적인 타임 아웃 값을 적용함으로써 공정성의 저하를 막을 수 있다.

Acker의 올바른 선택을 위해서 본 논문에서는 손실률과 RTT, 두 요소를 모두 사용하여 쓰루풋 비교를 통해서 NAK 억압을 정하게 한다. (그림 5)은 라우터에서의 쓰루풋 비교를 통하여 NAK 억압을 하는 과정을 순서대로 표시하였다. (그림 6)에서 보듯이 (그림 4)와 같이 NAK 억압을 하는 경우와는 달리 송신자측에 가까운 라우터에서 늦게 온 NAK라도 무조건적으로 억압을 하지 않고 그 전에 온 NAK의 쓰루풋과 늦게 온 NAK에 담긴 정보를 가지고 쓰루풋 비교를 통하여 억압을 하는지 결정한 후 만약 늦게 온 NAK의 쓰루풋이 더 낮은 경우 송신자에게 전달하여 보다 정확한 Acker 선택을 할 수 있게 하고, 불필요한 Acker 변경을 막을 수 있다.



(그림 5) 쓰루풋 비교 순서도



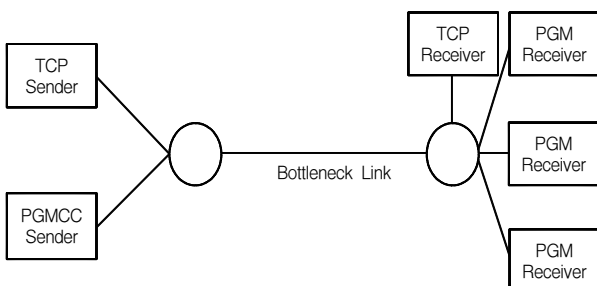
(그림 6) 쓰루풋 비교에 의한 Acker의 올바른 선택

4. 실험결과 및 분석

본 논문에서는 PGMCC에 위와 같은 새로운 방법을 적용하기 위하여 버클리 대학의 NS-2를 이용하여 실제 네트워크망을 시뮬레이션 하였다.

4.1 적응적인 타임 아웃 값의 적용

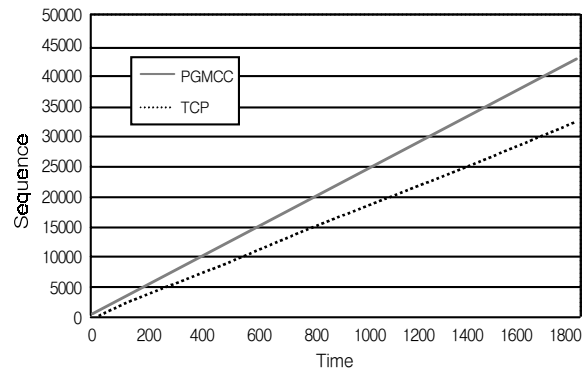
본 실험의 시뮬레이션 토폴로지는 (그림 7)과 같다.



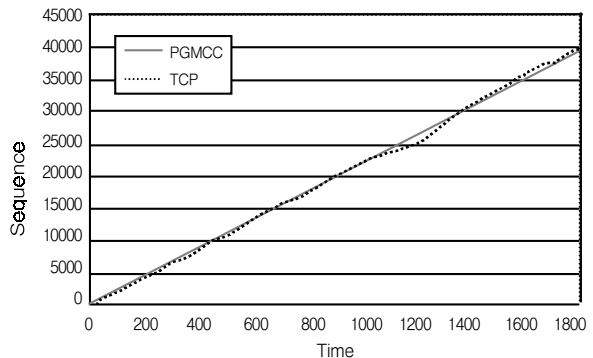
(그림 7) 시뮬레이션 토폴로지

병목구간의 대역폭은 500kb/s이고 지연시간은 50ms이다.

(그림 8)과 (그림 9)는 기존의 PGMCC와 적응적인 타임아웃 값을 적용한 PGMCC와 TCP와의 공정성 비교를 나타내었다. 실험결과에서 볼 수 있듯이 기존의 PGMCC는 시간이 지날수록 공정성이 심하게 훼손됨을 알 수 있으나, 적응적인 타임아웃을 적용하였을 때의 PGMCC와 TCP는 나란히 비슷한 추세를 나타내며 공정성을 보여주고 있다.



(그림 8) 적응적인 타임아웃을 적용하지 않은 PGMCC와 TCP와의 공정성



(그림 9) 적응적인 타임아웃을 적용한 PGMCC와 TCP와의 공정성 비교

이를 수치로 나타내기 위해서 다음과 같은 공정성 척도를 나타내어보았다. 공정성 수치는 [6]의 연구결과에서 인용하였다.

$$fairness\ index = \frac{(\sum x_i)^2}{n \sum x^2} \tag{3}$$

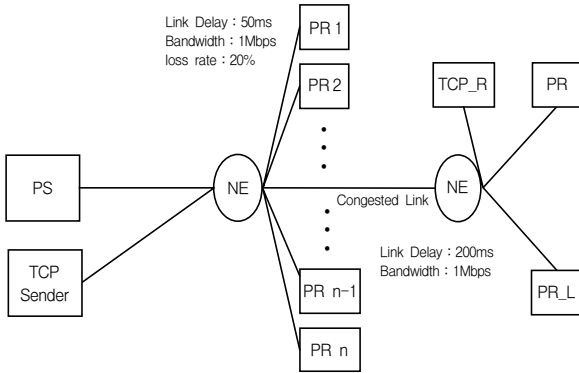
$x_i$ 는 병목구간을 공유하는 연결들이 공평하게 나누어 쓸 때의 대역폭으로서 각 연결의 전송률을 정규화한 값이다. <표 1>은 (그림 8)과 (그림 9)의 결과를 공정성 수치를 사용하여 나타낸 것이다.

<표 1> 공정성 수치 비교

PGMCC	공정성 수치(Fairness Index)
기존 PGMCC	0.71
제안된 방법을 적용한 PGMCC	0.98

4.2 Acker의 올바른 선택

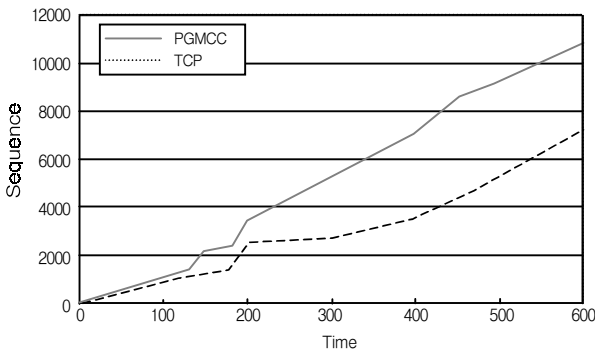
적응적인 타임 아웃 값을 적용시켜 공정성을 향상시킨 상태에서 (그림 10)과 같은 시뮬레이션 토폴로지에서 다시 한번 공정성을 측정해보았다.



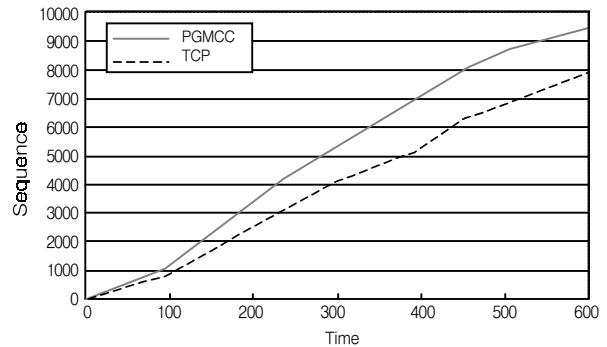
(그림 10) 시뮬레이션 토폴로지

공정성을 개선하기 위해서 새로운 Acker 선택 방식을 적용시켜 얼마나 빠르게 올바른 Acker가 선출되는지를 비교 실험해 보았다. 병목구간의 대역폭은 1Mb/s이고, 지연시간은 50ms이다. PR은 모두 PGMCC 수신자로서 PR 1~n은 20% 패킷손실률을, PR\_L은 25%의 패킷 손실률을 가진다.

우선 위의 토폴로지에서 n = 6(n은 수신자의 수)일때의 적응적인 타임 아웃 값을 적용시킨 기존의 PGMCC와 새로운 Acker 선출방식까지 적용한 PGMCC와 TCP와의 공정성을 비교해 본 결과 (그림 11)과 (그림 12)와 같은 결과를 얻을 수 있었다. 적응적인 타임 아웃 값을 적용시킨 기존의 PGMCC에서는 쓰루풋 비교를 통해서 약 3분에서 7분 사이에 불확실한 Acker의 선정 때문에 공정성이 저하되는 것을 (그림 11)에서 볼 수 있다. 반면에 (그림 12)는 새로운 Acker 선출방식까지 적용한 PGMCC에서의 쓰루풋 비교를 통한 결과를 보여주고 있으며 (그림 11)에 비해 공정성이 개선되었음을 직관적으로 확인할 수 있다. 이때의 공정성 수치는 0.82로 (그림 11)의 0.57보다 공정성이 개선됨을 공정성 수치로도 알 수 있다(식 (3)).



(그림 11) NAK 억압에 의하여 공정성이 저하된 PGMCC와 TCP의 쓰루풋 비교



(그림 12) 공정성이 개선된 PGMCC와 TCP의 쓰루풋 비교

다음으로 n을 2부터 8까지 증가시키면서 기존의 PGMCC의 방법과 새롭게 제안된 Acker 선택 방법을 모두 적용시켜서 Acker가 올바르게 선정되는 시간에 따른 Acker의 수를 측정하였다. 각 n의 경우마다 50회씩 시뮬레이션을 행하였고 그 결과를 <표 2>에 확률분포로 나타내었다.

<표 2> 올바른 Acker 선택 시간에 따른 확률분포

Receiver의 숫자	1분 이내		1~10분		10분 이후	
	기존의 방법(%)	새로운 방법(%)	기존의 방법(%)	새로운 방법(%)	기존의 방법(%)	새로운 방법(%)
2	55	96	43	4	2	•
4	44	88	48	12	8	•
6	38	88	48	12	14	•
8	36	89	40	11	24	•

실험결과와 같이 n = 2일 경우에 기존의 방법의 경우 1분 이내에 올바른 Acker로 선정되는 경우는 약 50%이며, n이 커질수록 1분 이내에 올바른 Acker로 선정되는 확률이 현저히 줄어든다. 뿐만아니라 10분 이후에 올바른 Acker가 선정되어지는 확률이 높아진다. 즉, 기존의 방법은 올바른 Acker를 선택함에 있어서 10분 이상 긴 시간이 걸리는 경우가 발생할 수 있다. 하지만 새로운 방법을 적용한 결과, 수신자의 수의 증가와는 상관없이 일관적으로 올바른 Acker 선택 시간이 1분 이내로 줄어드는 확률이 높아진다. 만약 1분 이내에 올바른 Acker가 선정이 되지 않는다 할지라도 10분 이내에는 올바른 Acker가 100% 선정된다. 그러므로 기존의 방식에 비해 정확한 Acker의 선정에 걸리는 시간이 줄어들기 때문에 공정성에 있어서도 향상을 가져오고 있음을 보여주고 있다.

4. 결 론

PGMCC는 NAK 억압 방식을 사용하여 쉼의 폭주 현상을 방지하고 수신자 중에 Acker를 선정하여 TCP와의 공정성을 추구했다. 그러나, 고정된 타임 아웃 값의 사용과 Acker를 선택함에 있어서 불확실함은 TCP와의 공정성의 저하를 가져 올 수 있다. 이러한 문제를 해결하기 위해서 본 논문

에서는 PGMCC에 적응적인 타임 아웃 값을 적용하고, 라우터의 지원을 통한 쓰루풋 비교로 NAK 억압을 하는 방식을 채택하였다. 시뮬레이션을 통하여 본 논문에서 적용한 방법들이 PGMCC에서의 공정성을 개선하는 것을 보여주었다.

추후 과제로는 시뮬레이션이 아닌 실제 네트워크망에서의 실험과 Acker 선택에 있어서 불확실한 점을 보완하기 위하여 본 논문에서 지적한 다른 두 가지 방법을 적용시키는 것이 있다.

### 참 고 문 헌

[1] Chwan-Hwa Wu, J. David Irwin, "Emerging Multimedia Computer Communication Technologies," Prentice Hall, 1998.  
 [2] L. Rizzo, L. Vicisano, M. Handley and G. Iannaccone, "PGMCC single rate multicast congestion control : protocol specification," Internet Draft IETF, Feb., 2001.  
 [3] Tony Speakman, Dino Farinacci, Jon Crowcroft, Dan Leishchiner, Luigi rizzo, PGM Reliable Transport Protocol Specification, Internet-Draft, draft-speakman-pgm-spec-05.txt, November, 2000.  
 [4] Chin-ying Wang and Sonia Fahmy, "Dynamics of the "pgmcc" Multicast Congestion Control protocol," Tektronix Report, August, 2001.  
 [5] W. Richard Stevens, "TCP/IP Illustrated, Volume I," Addison-Wesley, 1994.  
 [6] R. Jain, A. Duresi, "Throughput Fairness Index : An Explanation," ATM Forum/99-0045, Feb., 1999.  
 [7] J. Padhye, V. Firoiu and J. Kurose, Modeling TCP throughput : A simple Model and its Empirical Validation, ACM SIGCOMM 1998, Vancouver, BC, Canada, Sepetember, 1998.  
 [8] Luigi Rizzo, "pgmcc : a TCP-friendly single-rate multicast congestion control scheme," ACM SIGCOMM00, 2000.  
 [9] S. Floyd, M. Handley, J. Padhye and J. Widmer, "Equation-based congestion control for unicast Applications," Proc. ACM SIGCOMM, pp.43-56, Aug., 2000.

[10] I. Rhee, V. Ozdemir and Y. Yi, "TEAR : Tcp emulation at receivers-flow control for multimedia streaming," Technical report, Department of Computer Science, Apr., 2000.  
 [11] Sneha K. Kasera, Supratik Bhattacharyya, Mark Keaton, "Scalable Fair Reliable Multicast Using Active Services," IEEE/ACM Transactions on Networking, Aug., 2000.



### 박 영 선

e-mail : hayame@lge.com

2001년 서강대학교 전자공학과(학사)

2003년 서강대학교 대학원 전자공학과  
(공학 석사)

2003년~현재 LG전자 Digital & Multimedia  
연구소 연구원

관심분야 : 멀티캐스트, 고속통신망, Multimedia Messaging



### 현 도 원

e-mail : snatcher@eecal.sogang.ac.kr

2003년 서강대학교 전자공학과(학사)

2003년~현재 서강대학교 대학원 전자공학과  
석사과정

관심분야 : 인터넷 프로토콜, 멀티캐스트,  
라우팅 아키텍처



### 장 주 욱

e-mail : jjang@mail.sogang.ac.kr

1983년 서울대학교 전자공학과(학사)

1985년 한국과학기술원 전기 및 전자공학과  
(공학석사)

1993년 University of Southern California  
컴퓨터공학과(공학박사)

1995년~현재 서강대학교 전자공학과 부교수

관심분야 : 인터넷 프로토콜, 병렬처리