

PGMCC의 공정성 향상

박형선, 장주욱
서강대학교 전자공학과

hayame@eece1.sogang.ac.kr, jjang@mail.sogang.ac.kr

Enhancing the Fairness of the PGMCC
Young-sun, Park, Ju-wook Jang
Dept. of Electronic Engineering, Sogang University

요약

본 논문은 멀티캐스트 혼잡제어방식 제안 중의 하나인 PGMCC의 공정성 향상 방법에 관한 것이다. 안정적인 멀티캐스트 전송과 혼잡제어를 위한 여러가지 방법중에서 PGMCC는 유력한 방식으로 제안되었다. 하지만 PGMCC에도 고정된 timeout 값의 사용과 불확실한 acker 선출로 인하여 공정성을 낮추는 요인들이 몇가지 지적되고 있다. 본 논문에서는 TCP와 같은 적응적인 timeout값의 적용과 acker 선출에서 추가적인 방법의 적용으로 이와 같은 문제를 해결하여 공정성이 향상되는 것을 확인 할 수 있었다.

1. 서론

멀티캐스트는 다수의 사용자에게 동일한 데이터를 전송할 때 하나의 링크에 동일한 데이터 스트림이 두번 이상 지나가지 않게 함으로서 대역폭을 절감하는 전송 방법으로서 다수에게 멀티미디어 전송을 하는 VOD, 실시간 인터넷 원격강의 시스템과 같은 경우에 데이터의 양을 획기적으로 줄일 수 있게 한다.[1] 그러나 기존에 널리 사용되고 있는 TCP의 경우에는 혼잡제어가 되는 반면 멀티캐스트의 경우에는 혼잡제어가 이루어지지 않기 때문에 같은 인터넷망에서 공존할 수가 없다. 이러한 문제점을 해결하기 위하여 rmtp, tmcc, pgm등의 여러가지 멀티캐스트 방식들이 제안되고 있다. 이런 방식중에서 PGMCC[4]는 단일 속도 제어의 대표자 기반 멀티캐스트 혼잡제어방식이다. PGMCC는 Acker라 불리는 대표자와의 데이터 교환을 통해 전송 윈도우의 크기를 조절한다. acker는 수신자중에서 가장 낮은 전송율을 가지는 수신자가 acker로 선출된다. PGMCC는 현재 NACK중심의 멀티캐스트 혼잡제어방식중에서 유망한 제안 중의 하나지만 몇가지의 단점을 가지고 있다. PGMCC의 경우 timeout값이 일정하기 때문에 선택된 Acker의 rtt에 따라서 공정성이 심각하게 손상될 수 있다. 또한 acker의 선택에 있어서 NACK 억압에 의해서 올바른 acker가 선택되지 않을 수 있다. 본 논문에서는 이러한 단점을 보완하기 위해서 TCP와 같은 adaptive timeout방식을 적용하였고, acker 선택방법을 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 추후연구방향에 대해서 서술하였다.

2. 기존 PGMCC의 문제점

2.1 PGMCC(PGM Congestion Control)

PGM(Pragmatic Genral Multicast)[3]은 단일 송신자 방식의 멀티캐스트 방식으로 NAK를 기반으로 한 저전송 요청을 통하여 안정적인 멀티캐스트를 제공한다. PGM에서는 궤환 억압을 통하여 멀티캐스트망의 확장성을 제공한다.

그림 1은 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 메시지를 전달하게 된다. 만약에 같은 NAK를 이 라우터가 받았을 경우에는 NAK를 전송하지 않고 억압하게 된다.

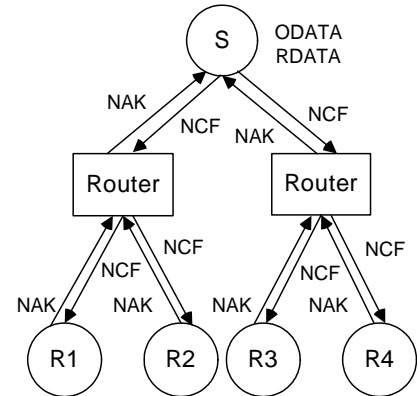


그림 1 PGM 시나리오

PGMCC[4]는 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. 간단한 시나리오를 그림 2에서 확인 할 수 있다. acker는 그룹의 대표자로서 가장 나쁜 throughput을 가지는 수신자가 선택된다. 송신자와 acker 사이의 전송률은 TCP와 비슷한 window 기반으로 결정되며, 이는 역시 ack에 의하여 결정된다. pgmcc에서 수신자가 throughput을 판단하는 것은 수신자의 보고에 의해서 결정된다. 수신자는 nack나 ack에 들어있는 loss rate와 sequence number를 이용하여 RTT를 측정 throughput을 결정하게 된다. loss rate는 전송을 피하기 위하여 저대역필터를 사용하여 계산을 한다.

pgmcc에서 가장 중요한 점은 acker의 선택에 있다. acker는 가장 낮은 throughput을 가지는 수신자가 선택된다. Acker의 선택은 Throughput의 비교를 통해서 결정한다.

$$T \propto \frac{1}{RTT \sqrt{p}} \quad (1)$$

T : throughput
RTT : round trip time
p : Loss rate

이를 계산하여 기존의 대표자 i 와 수신자 j 로부터 받은 nack을 기반

으로 하여 $T(i) > T(j)$ 일 경우 수신자 j 에서 수신자 i 로 대표자가 바뀌게 된다. 그림 2는 rtt 측정에 대한 그림이다.

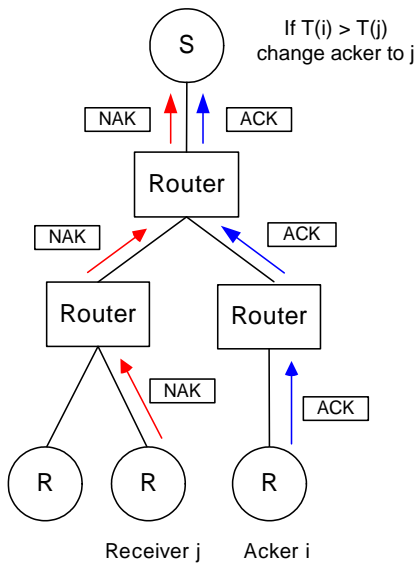


그림 2 PGMCC 시나리오

2.2 문제점

PGMCC는 timeout값을 정하는 데에 있어 TCP와는 다르게 특별한 방법을 두지 않고 처음에 정한 값으로 계속 timeout값을 유지하게 된다. PGMCC에서 고정적인 timeout값을 사용하는 이유는 PGMCC의 경우 rtt 의 측정을 위하여 timestamp를 사용하는 것이 아니라 sequence number를 사용하기 때문이다. PGMCC에서 rtt 는 단지 대표자를 선택할 때, 수신자들간의 throughput 비교를 위해 사용되기 때문에 sequence number를 사용해서 rtt 를 측정하는 것이 더 낫다고 제안하고 있다. 하지만 이런 방법은 timeout에 필요한 정확한 rtt 를 산출해 낼 수 없기 때문에, TCP와 같은 적응적인 timeout값을 산출해 낼 수 없다. TCP와 같은 윈도우 방식을 사용하는 PGMCC의 경우에 timeout 값을 어떻게 정하느냐에 따라 공정성에 큰 영향을 미치게 된다. 만약, acker의 rtt 가 timeout 값보다 크게 작게 정해지면은 PGMCC의 경우 timeout이 잘 안일어나기 때문에 TCP보다 윈도우 사이즈가 1로 잘 줄어들지 않게 되기 때문에 TCP는 상대적으로 낮은 throughput을 가지게 된다. 반대로, acker의 rtt 가 timeout값보다 크게 되면은 계속해서 timeout이 발생하기 때문에 전송 윈도우의 사이즈가 1이 되어 데이터를 제대로 전송하지 못하게 된다. 실제 네트워크의 경우 네트워크의 혼잡도는 다양하게 변하기 때문에 이런 고정적인 timeout값은 공정성에 큰 영향을 미칠 수가 있다.[5]

다음은 PGMCC 논문에서도 중요한 문제로 지적했듯이 올바른 acker의 선택이다. PGMCC는 가장 낮은 throughput을 가지는 수신자를 대표자로 선택함으로써 TCP와 친화적인 공정성을 보장할 수 있다. PGM에서 제안된 제한 역압 방식은 가장 먼저 라우터에 도착한 NAK를 송신자에게 올리 보내고 다음에 도착하는 NAK의 경우 송신자에게 전송하지 않는다. 이 때, 비록 rtt 는 크지만 패킷 손실율이 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 throughput은 알 수가 없으므로, 대표자 선택에서 제외되게 된다. 잘못된 Acker를 선정할 경우 공정성의 저하가 올 수 있다.

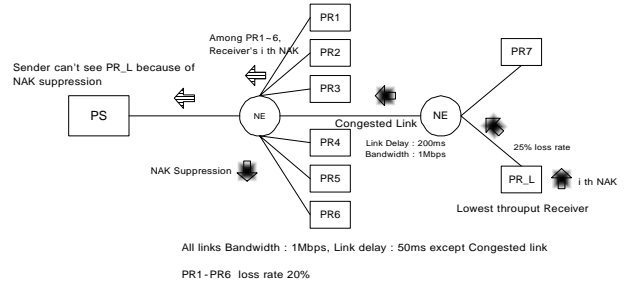


그림 3 NAK Suppression에 의한 Acker의 잘못된 선택

3. 제안된 방법

고정적인 timeout값에 의한 공정성 저하는 TCP와 같은 적응적인 timeout값을 적용함으로써 해결할 수 있다. 이 경우 기존의 PGMCC 논문에서 제안한 Sequence number를 사용한 rtt 값이 아닌 timestamp를 사용한 rtt 를 사용해야 한다. timestamp를 사용할 경우 TCP와 마찬가지로 timeout값을 산출해 낼 수 있다. timeout값이 고정적으로 정해져 있는 PGMCC에 적응적인 timeout값을 적용함으로써 공정성의 저하를 막을 수 있다.

Acker의 올바른 선택을 위해서 본 논문에서는 loss rate와 rtt , 두 요소를 모두 사용하여 throughput 비교를 통해서 NAK 역압을 정확하게 한다. 그림 4과 같이 먼저 NAK j 가 라우터에 통하여 상단에 먼저 전달되고 늦게 도착한 NAK라도 라우터에서의 throughput 비교를 통하여 역압되지 않고, 송신자에게 전달되어 보다 정확한 Acker 선택을 할 수 있게 된다.

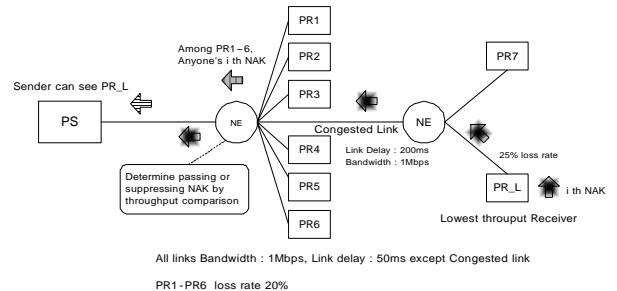


그림 4 Throughput Comparison에 의한 Acker의 올바른 선택

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 timeout값의 적용

본 실험의 시뮬레이션 보플로지는 그림 5와 같다.

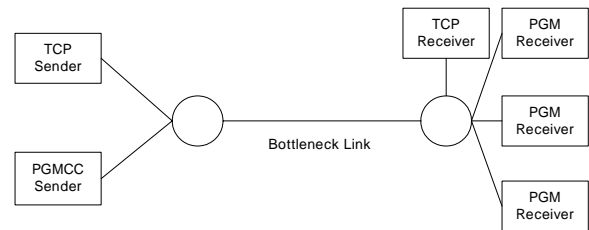


그림 5 시뮬레이션 보플로지

병목구간의 대역폭은 500kb/s이고 지연시각은 50ms이다. 그림 5과 6은 기존의 PGMCC와 적응적인 timeout값을 적용한 PGMCC와 TCP와의 공정성 비교이다.

실험결과에서 볼 수 있듯이 적응적인 timeout을 적용하였을 때의 P

GMCC와 TCP는 나란히 비슷한 추세를 나타내며 공정성을 보여주고 있다. 이를 수치로 나타내기 위해서 다음과 같은 공정성 척도를 나타내어 보았다. 공정성 수치는 [6]의 연구결과에서 인용하였다.

표 1 공정성 비교

PGMCC	공정성(Fairness)
고정적인 timeout값 적용	0.71
적응적인 timeout값 적용	0.98

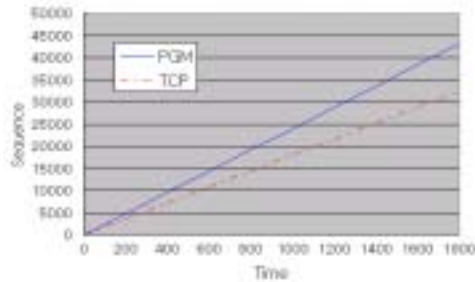


그림 6 적응적인 timeout을 적용하지 않은 PGMCC와 TCP와의 공정성

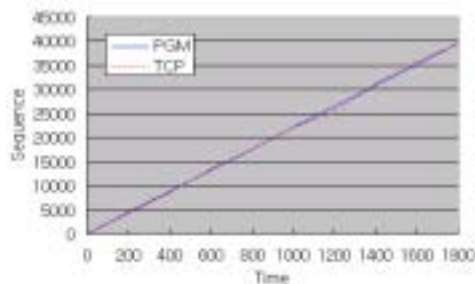


그림 7 적응적인 timeout을 적용한 PGMCC와 TCP와의 공정성 비교

4.2 Acker의 올바른 선택

본 실험의 시뮬레이션 보물표지는 그림 9와 같다.

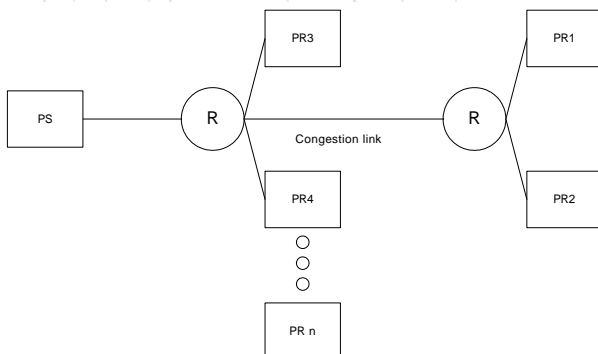


그림 8 시뮬레이션 보물표지

병목구간의 대역폭은 10Mb/s이고, 지연시간은 5ms이다. PR은 모두 PGMCC 수신자로서 PR 1,2는 20%대역폭을 할당, PR3,4,5..n는 25%의 대역폭을 가진다. 실험은 n을 8까지 증가시키면서 기존의 PGMCC의 방법과 새롭게 제안된 Acker 선택 방법을 모두 적용시켜서 acker가 올바르게 선정되는 시간을 측정하여 보았다. 각 n의 경우마다 50회씩

시뮬레이션을 행하였다. 그 결과는 표 2와 같다.

표 2 실험 결과

Receiver의 숫자	1분 이내		1~10분		10분 이후	
	기존의 방법(%)	새로운 방법(%)	기존의 방법(%)	새로운 방법(%)	기존의 방법(%)	새로운 방법(%)
2	56	96	43	4	2	•
4	44	88	48	12	8	•
6	38	88	48	12	14	•
8	36	89	40	11	24	•

실험결과와 같이 혼잡링크 전단의 수신자들의 수를 증가시킴에 따라 기존의 방법의 경우 10분 이후에 올바른 Acker가 선택되어지는 경우가 많아지게 된다. 하지만 새로운 방법을 적용한 결과 수신자의 수의 증가와는 상관없이 일관적으로 1분 이내에 올바른 Acker가 결정되어 지는 경우가 많아진다.

4. 결론

PGMCC는 NAK 역압 방식을 사용하여 채환의 폭주 현상을 방지하고 수신자 중에 acker를 선정하여 TCP와의 공정성을 추구하였다. 그러나, 고정된 timeout값의 사용과 acker를 선택함에 있어서 불확실함은 TCP와의 공정성의 지해를 가져 올 수 있다. 이러한 문제를 해결하기 위해서 본 논문에서는 PGMCC에 적응적인 timeout값을 적용하고, 라우터의 지령을 통한 throughput 비교로 NAK 역압을 하는 방식을 채택하였다. 시뮬레이션을 통하여 본 논문에서 적용한 방법들이 PGMCC에서의 공정성을 개선하는 것을 보여주었다.

후주 과제로는 시뮬레이션이 아닌 실제 네트워크망에서의 실험과 acker 선택에 있어서 불확실한 점을 보완하기 위하여 본 논문에서 지적한 다른 두가지 방법을 적용시키는 것이 있다.

참고문헌

- [1] Chwan-Hwa Wu, J. David Irwin, "Emerging Multimedia Computer Communication Technologies", Prentice Hall, 1998
- [2] J.Padhye, V.Firoiu, and J.Kurose. Modeling TCP throughput : A simple Model and its Empirical Validation. ACM SIGCOMM 1998, Vancouver, BC, Canada, September 1998
- [3] Tony Speakman, Dino Farinacci, Jon Crowcroft, Dan Leshchiner, Luigi rizzo. PGM Reliable Transport Protocol Specification. Internet-Draft, draft-speakman-pgm-spec-05.txt, 24 November 2000.
- [4] L.Rizzo, L.Vicisano, M.Handley, and G.Ianaccone, "PGMCC single rate multicast congestion control : protocol specification," Internet Draft IETF Feb. 2001
- [5] Chin-ying Wang and Sonia Fahmy "Dynamics of the "pgmcc" Multicast Congestion Control protocol" Tektronix Report, August 2001
- [6] R.jain, A.Durresi "Throughput Fairness Index : An Explanation".ATM Forum/99-0045, Feb. 1999
- [7] Luigi Rizzo "pgmcc : a TCP-friendly single-rate multicast congestion control scheme", ACM SIGCOMM00, 2000