

Active Network 기술을 이용한 신뢰성 있는 멀티캐스트의 혼잡 제어

*이종권, *이상정, *김탁곤, **최길영, **이규호
*한국과학기술원 전자전산학과 전기 및 전자공학전공
**한국전자통신연구원 라우터기술연구부

Reliable Multicast Congestion Control using Active Networking

*Jong-Kwon Lee, *sangjeong Lee, *Tag Gon Kim, **Kil Young Choi, and **Kyou Ho Lee
*Division of Electrical Engineering, Dept. of EECS, KAIST
**Router Technology Dept., ETRI

요 약

본 논문에서는 신뢰성 있는 멀티캐스트의 혼잡 제어를 위한 알고리즘을 제안하고, 이를 구현하는 방안으로서 Active Network 기술을 이용하는 방법을 제안하고 있다. 멀티캐스트에서의 혼잡 제어 위해 일대일 통신에서 제안된 credit-based hop-by-hop 흐름 제어 방법을 확장하였으며, 각 네트워크 노드에서 버퍼의 오버플로우를 방지하여 흐름 제어뿐만 아니라 데이터 전송의 신뢰성도 높일 있게 된다. 또한, 네트워크 노드에서 제공되어야 하는 이러한 기능들을 구현하는 방안으로서 Active Network 기술을 이용하는 방법에 대해 제안하고 있으며, 이산 사건 시뮬레이션을 통하여 제안된 알고리즘의 성능 분석을 행하였다.

1. 서론

멀티캐스트 혼잡 제어 문제는 매우 어려운 문제로 여겨지고 있으며, 특히 신뢰성 있는 멀티캐스트의 경우에는 최근에서야 몇 가지 제어 방법들이 제안되고 있다. 신뢰성 있는 멀티캐스트 흐름 제어 문제의 주된 이슈로는 피드백 메커니즘의 확장성(scalability) 문제와, 일대일통신 트래픽과 멀티캐스트 트래픽 사이의 공정성(fairness) 문제를 꼽을 수 있다. 대부분의 제안된 방법들은 종단간(end-to-end)에 작용하는 방법들이며, 확장성과 공정성 문제에 대한 완벽한 해법을 제시하지 못하고 있다[1][2].

본 논문에서는 멀티캐스트 통신을 위한 credit-based hop-by-hop 흐름 제어를 제안하고 있다. 제안된 방법은 일대일 통신에서 제안된 credit-based 흐름 제어 메커니즘을 멀티캐스트의 경우로 확장한 것이다. 이 메커니즘을 사용하면 각 노드에서 버퍼의 오버플로우를 방지하여 데이터 전송의 신뢰성을 높이고 버퍼를 효율적으로 관리할 수 있으며, 버퍼에 수용될 수 있는 정도의 조절된 트래픽만이 노드에 들어오게 된다. 한편, 이와 비슷하게 다중계층으로 부호화된 비디오 트래픽에 적용한 연구가 있었지만, 데이터의 손실을 허용하지 않는 신뢰성 있는 멀티캐스트에는 적용할 수 없었다[3].

제안된 혼잡 제어 방법은 네트워크 내에서 구현되어 작용하게 되는데, 멀티캐스트 흐름 제어를 네트워크 내에서 하게 될 때 얻어지는 몇 가지 장점이 있다. 먼저 응답성 측면에서는 종단가(end-to-end)의 혼잡 제어에 비해 네트워크 상태에 보다 신속히 반응할 수 있다는 점이다. 확장성 측면에서 보면, hop-by-hop 흐름 제어는 멀티캐스트 연결을 바로 아래쪽 노드에 대한 다중의 일대일 연결로 나누어 처리하기 때문에 네트워크 상태에 대한 피드백 정보들이 모든 노드들에 골고루 나뉘어져서 확장성 문제를 해결하고 있다. credit-based 흐름 제어에서 교착상태(deadlock)에 빠지는 것을 방지하기 위해 연결별 대기(per-connection queuing) 방법을 사용해야 하는데, 이것은 동시에 일대일 통신 트래픽 또는 다른 멀티캐스트 트래픽과의 공정성을 제공하는 방법이 될 수 있다.

제안된 혼잡 제어를 사용하기 위해서는 모든 노드에서 구현되어야 한다는 점이 단점으로 지적되고 있다. 하지만, 최근에 등장하고 있는 active network[4]은 향후 네트워크의 진화 방향을 제시하고 있으며, 이러한 active network 기술을 이용하면, 제안된 멀티캐스트 혼잡 제어 방법이 효과적으로 실행될 수 있을 것으로 생각된다.

본 논문의 구성은 다음과 같다. 2 절에서는 멀티캐스트 혼잡 제어를 위한 credit-based 흐름 제어 알고리즘에 대해 자세히 설명한다. 제안된 혼잡 제어 방법을 구현하기 위해 active network 기술을 이용하는 방안에 대해서 3 절에서 설명을 하며, 4 절에서는 이산 사건 시뮬레이션 환경인 DEVSim++ 을 통해 제안된 혼잡 제어 방법의 성능 평가를 하고 있다. 마지막으로 5 절에서 결론을 맺고 있다.

2. 신뢰성 있는 멀티캐스트를 위한 Credit-based 흐름 제어 알고리즘

본 논문에서 제안하는 멀티캐스트 혼잡 제어 알고리즘은 ATM 네트워크에서 일대일 통신에 대하여 제안된 credit-based 흐름 제어 알고리즘을 멀티캐스트 연결에 대하여 확장한 것이다. 일대일 통신에서의 credit-based 흐름 제어를 요약하면 다음과 같다[5]. 즉, 송신자와 수신자 간의 연결 경로에 있는 이웃한 두 노드에 대하여:

- 아래쪽 노드

- M 개의 패킷을 전송할 때마다 아래쪽 노드의 버퍼에서 수용 가능한 패킷의 양을 credit 패킷에 담아 위쪽 노드로 송신한다.

- 위쪽 노드

- Credit 이 남아 있는한 데이터 패킷을 아래쪽 노드로 계속 전송한다.

- 아래쪽 노드로부터 credit 패킷을 수신하면, credit을 갱신한다.

멀티캐스트 연결에 대해서 확장하면 이상과 같은 credit 갱신(update) 알고리즘이 약간 수정된다.

● 아래쪽 노드

- 멀티캐스트 연결에 해당하는 모든 출력 포트에서 최소한 M 개의 패킷을 전송할 때마다 credit 패킷을 위쪽 노드로 송신한다.

● 위쪽 노드

- 멀티캐스트 연결의 각 출력 포트는 각자의 credit 이 남아 있는한 복사된 데이터 패킷을 각 출력 포트의 아래쪽 노드로 계속 전송한다.

- 각 출력 포트의 아래쪽 노드로부터 credit 패킷을 수신하면, 해당하는 출력 포트의 credit 을 갱신한다.

2.1 Credit 갱신 알고리즘

Credit 갱신 알고리즘에서 사용되는 변수들은 표 1과 같다.

표 1: 멀티캐스트 연결을 위한 Credit 갱신 알고리즘에서 사용되는 변수들

$B_{i,p}$	Buffer allocation for MC i at port $p \in P_i$
$FC_{i,p}$	Forward counter for MC i at port $p \in P_i$
$TC_{i,p}$	Transmit Counter for MC i at port $p \in P_i$
$CB_{i,p}$	Credit balance for MC i at port $p \in P_i$
P_i	A set of ports through which packets of MC i go out
M	Credit update period in packets

MC: Multicast Connection

앞에서도 설명한 것처럼, 멀티캐스트 연결(multicast connection, MC) i 의 아래쪽 노드는 MC i 의 각 출력 포트에서 최소한 M 개의 데이터 패킷을 전송할 때마다 버퍼할당 $B_i(D)$ 와 포워드 카운터 $FC_i(D)$ 를 바로 위쪽 노드로 보낸다.

위쪽 노드의 출력 포트 p 에서는 수신된 credit 패킷으로부터 새로운 Credit Balance $CB_{i,p}(U)$ 를 다음과 같은 식에 의해 다시 계산한다.

$$CB_{i,p}(U) = B_i(D) + FC_i(D) - TC_{i,p}(U).$$

위쪽 노드에서는 데이터 패킷 전송시마다 $CB_{i,p}(U)$ 값은 하나씩 줄어들며, $CB_{i,p}(U)$ 이 0 이 되면 더 이상 데이터 패킷을 전송하지 않고 새로운 credit 패킷에 의해 $CB_{i,p}(U)$ 를 갱신하여 0 보다 클 때까지 기다린다.

아래쪽 노드에서 credit 패킷을 위쪽 노드로 보낼 때 포함하는 정보인 버퍼할당 $B_i(D)$ 와 포워드 카운터 $FC_i(D)$ 값은 MC i 의 각 출력포트로부터 다음과 같이 결정 된다:

$$B_i(D) = B_{i,k}, FC_i(D) = FC_{i,k}$$

여기서,

$$k = \arg \min_{p \in P_i} \{ B_{i,p} + F_{i,p} \}$$

가 된다.

그림 1 은 멀티캐스트 연결에 대한 credit-based 흐름 제어의 예제를 보여 주고 있다. 출력 포트 3 이 병목지점이 되며, 따라서 입력되는 트래픽 속도는 포트 3 의 서비스 속도에 맞춰 조절된다. 한편, 포트 1과 2에서는 포트 3에 비해 버퍼에 있는 패킷을 모두 전송하고 비어 있을 때가 종종 있게 된다.

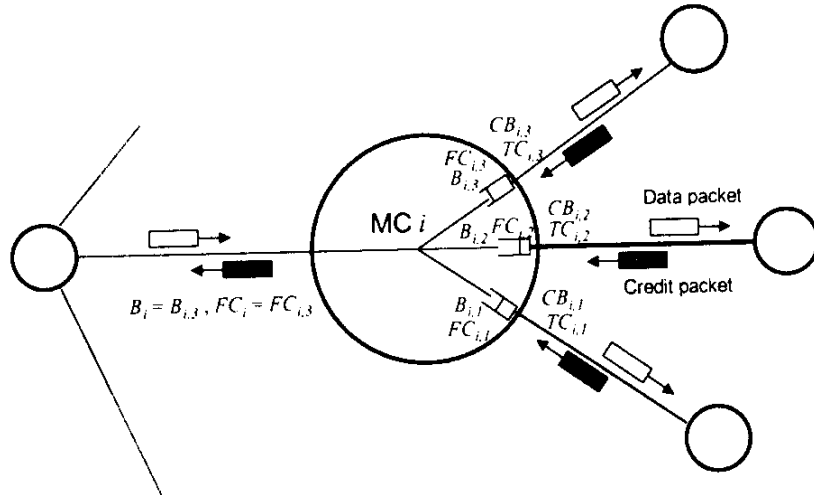


그림 1 예제: 멀티캐스트 연결에 대한 credit-based 흐름 제어

3. 멀티캐스트 혼잡 제어를 위한 Active Network 기술

2 절에서 제안된 credit-based 흐름 제어 방법은 여러 멀티캐스트 어플리케이션 중에서도 시간제한이 있고 신뢰성이 필요한 어플리케이션에 적합할 것으로 보인다. 한편, 제안된 메커니즘이 구현되기 위해서는 네트워크 내의 모든 노드에 구현되어야 하는데, 노드 내에서 수많은 연결에 대한 트래픽 상태를 관리하고 처리하는 것은 복잡도를 매우 높여서 또 다른 확장성 문제를 야기하게 된다. 따라서, 시간제한이 있고 신뢰성이 필요한 멀티캐스트 어플리케이션에 대해서만 실행되도록 하는 것이 바람직하다고 볼 수 있다.

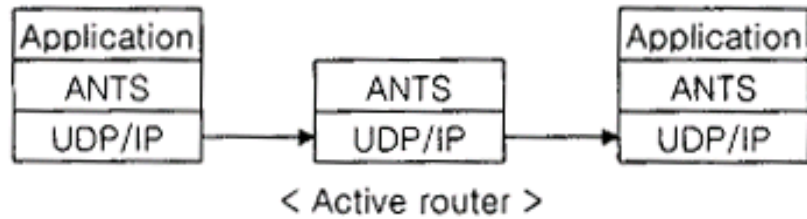
이를 위해서는 active network을 이용한 구현이 가장 적합하다. Active network 에서는 각 라우터들이 어플리케이션의 필요에 따른 능동 처리(Active Processing) 기능을 수행해 주어야 한다. MIT에서 개발된 ANTS(Active Network Transport System)[6]를 실행환경으로 사용하게 될 때의 active network 구조는 그림 2(가)와 같이 나타낼 수 있다.

제안된 멀티캐스트 흐름 제어 메커니즘을 사용하고자 하는 어플리케이션은 캡슐이라는 액티브 패킷을 보내어 라우터의 ANTS 실행환경에 의해 흐름 제어 프로그램이 설치하도록 한다. 그 이후에 전송되는 데이터 패킷들은 설치된 프로그램에 의해 각 노드에서 흐름 제어된다.

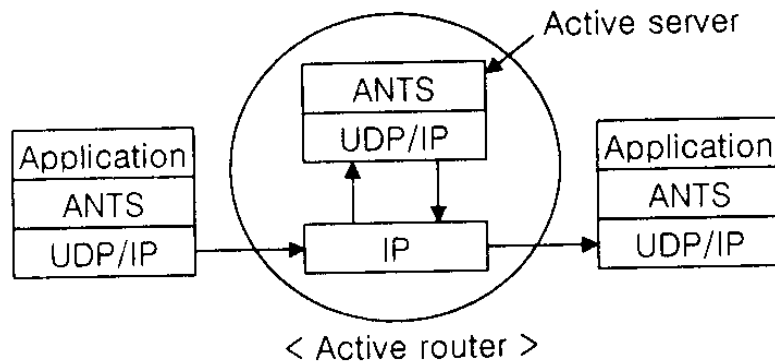
하지만, 현재 대부분의 라우터들은 이러한 능동처리 기능을 수용할 수 없는 구조이므로 그림 2(나)와 같은 형태로 가상적인 액티브 라우터를 구성할 수 있다. 즉, 능동처리 기능을 수행하게 되는 액티브 서버(active server)를 라우터에 부착하여 가상적으로 라우터에 능동처리 기능을 부여하도록 하는 것이다. 라우터에 들어오는 패킷 중 능동 처리가 필요한 액티

브 패킷을 액티브 서버로 전송하여 처리하게 한다.

액티브 서버를 부착하여 구현하면 현재 네트워크에 사용되고 있는 라우터들을 교체할 필요 없이 active network 기능을 구현할 수 있다. 하지만, 단점은 패킷의 처리 속도가 늦어진다 는 것이다. 그럼에도 불구하고, 현재의 네트워크에서 Active Network 으로 진화되는 과정 에서도 액티브 서버를 이용한 구현 방법이 이용될 수 있을 것이다.



(가) 능동 처리 기능을 직접 구현한 라우터



(나) 액티브 서버를 부착한 라우터

그림 2 Active Network 구조와 가상적인 구현방법

4. 시뮬레이션을 통한 성능 평가

제안된 멀티캐스트 흐름 제어에 대한 성능 평가를 위하여 이산사건 시뮬레이션 환경인 DEVSIM++ [7]을 이용하여 시뮬레이션을 수행하였다. 각 라우터의 출력 포트에서의 각 연결별 버퍼의 크기는 5 패킷으로 하였고, credit-based 흐름 제어 알고리즘에서 credit 패킷의 송신 주기 $M = 2$ 로 하였다.

먼저 그림 3 과 같은 네트워크 형태에 대하여 시뮬레이션을 수행하여, 제안된 알고리즘의 응답성과 일대일통신인 TCP 트래픽과의 공정성을 알아보았다. 송신측에서 보내고자 하는 데이터는 계속 공급된다고 가정하였고, 멀티캐스트 트래픽은 시간 0 에, TCP 트래픽은 5 초에 전송이 시작된다. 이때 TCP 트래픽도 credit-based 흐름 제어에 의해 전송률이 조정되는 것을 가정하였다. 모든 링크의 전파 지연시간은 5ms로 설정하였다.

이에 대한 시뮬레이션 결과는 그림 4 와 같다. 처음에 멀티캐스트 트래픽만 존재할 때는 병목 링크인 L4 의 대역폭을 모두 이용하는 것을 볼 수 있다. 이후에 TCP 트래픽이 교차하여 발생했을 때, 멀티캐스트 송신측에서의 전송률이 신속히 반응하여 TCP 트래픽과 거의 같은 대역폭을 나누어 가지는 결과를 확인할 수 있다.

제안된 알고리즘의 확장성에 관해 알아보기 위해 그림 3 의 라우터 Rt2 에 각각 1, 5, 10 개의 수신자가 있도록 하여 평균 전송률을 측정하였다. 그림 5 와 같이, 수신자의 크기가 커지더라도 평균 전송률은 거의 일정하게 유지됨을 알 수 있다.

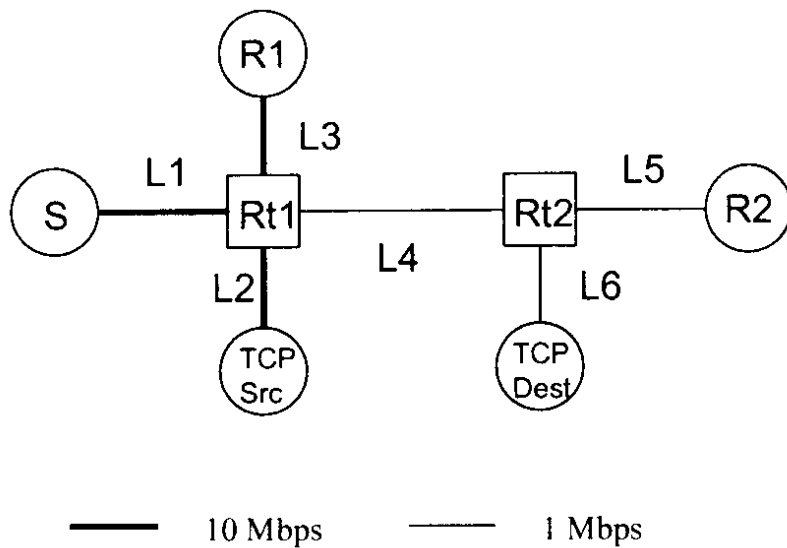


그림 3 시뮬레이션을 위한 네트워크 Topology

5. 결론

시뮬레이션 결과로부터, 제안된 알고리즘은 멀티캐스트 혼잡 제어의 주된 문제인 확장성과 공정성 문제를 잘 해결함을 알 수 있다. 또한, 네트워크 노드 사이에서 작용함으로써, 종단간의 혼잡 제어에 비해 네트워크 더 신속히 반응한다는 장점이 있다.

한편, 제안된 알고리즘을 구현하기 위해 active network 기술을 사용함으로써, credit-based 흐름 제어의 단점으로 지적되어 온 복잡도 문제를 어느 정도 해결할 수 있을 것으로 보인다.

추후 연구로는 제안된 알고리즘을 가상적인 액티브 라우터에 구현하는 한편, 종단간의 혼잡 제어와 결합하여 성능을 개선시킬 수 있는 방안을 연구하고자 한다.

참 고 문 헌

- [1] D. DeLucia and K. Obraczka, "Congestion Control Performance of a Reliable Multicast Protocol" Proc. Int. Conf. Network Protocols '98, pp.168-76.
- [2] I. Rhee, N. Balaguru, and G. N. Rouskas, "MTCP: Scalable TCP-like Congestion Control for Reliable Multicast," Proc. IEEE INFOCOM '99, pp.1265-73.
- [3] B. J. Vickers, C. Albuquerque, and T. Suda, "Adaptive Multicast of Multi-layered Video: Rate-Based and Credit-Based Approaches," Proc. IEEE INFOCOM '98, pp.1073-83.
- [4] D. L. Tennenhouse et al., "A Survey of Active Network Research," IEEE Communications Magazine, vol. 35, no. 1, Jan. 1997, pp.80-86.
- [5] H. T. Kung, T. Blackwell, and A. Chapman, "Credit-Based Flow Control for ATM Networks: Credit Update Protocol, Adaptive Credit Allocation, and Statistical Multiplexing," Proc. ACM SIGCOMM, 1994.
- [6] D. Wetherall, J. Guttag, and D. Tennenhouse, "ANTS: A Toolkit for Building and Dynamically Deploying Network Protocols," OPENARCH '98, April 1998.
- [7] T. G. Kim, DEVSIM++ User's Manual: C++ Based Simulation with Hierarchical, Modular DEVS Models, Systems Modeling Simulation Lab., KAIST, Taejon, Korea, 1994. <http://smsl.kaist.ac.kr/>.

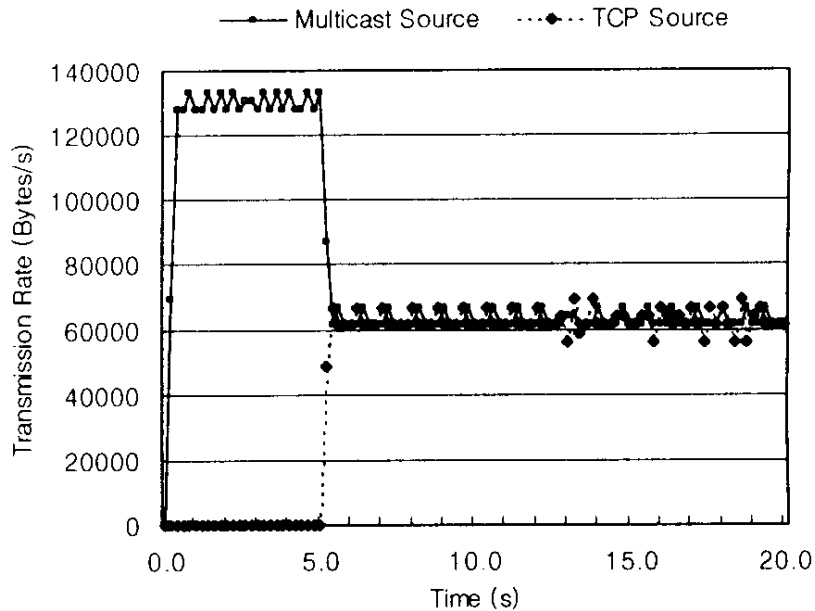


그림 4 멀티캐스트 트래픽과 TCP 트래픽

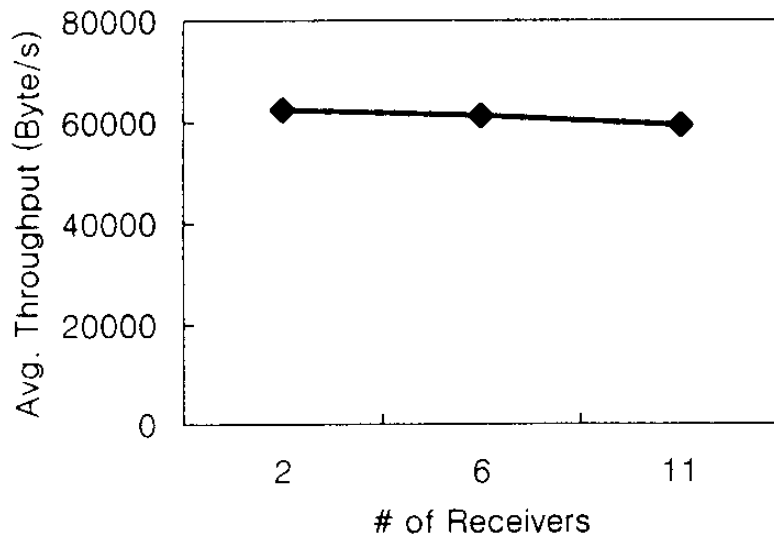


그림 5 수신자 수에 따른 평균 전송률의 변화