

인터넷상에서 공정성을 향상시키기 위한 멀티캐스트 혼잡 제어에 대한 이해

정인성*, 신용태

숭실대학교, 전자계산학과

Understanding Multicast Congestion Control to Improve Fairness in the Internet

Insung Jung*, Yongtae Shin

Dept. of Computer Science, Soongsil University
dormouse@cherry.soongsil.ac.kr, shin@computing.soongsil.ac.kr

Abstract

본 논문에서는 기존의 인터넷상에서의 멀티캐스트 프로토콜의 충돌 없는 전개를 위해 절대적으로 필요한 혼잡 제어의 주요 특징들을 집어 본다. 혼잡 제어가 반드시 고려되어야 할 사항은 성능(Throughput)과 공정성(Fairness)이다. 본 논문에서는 기존의 TCP기반 흐름과의 멀티캐스트 그룹간 흐름간의 상호 공정성(Inter-Fairness)과 멀티캐스트 그룹내의 흐름간의 공정성(Intra-Fairness)을 지원하기 위해 수반되는 문제점들에 대해서 이해한다. 전자의 경우, 흔히 *TCP-friendly 제어*라 하여 많이 연구가 진행중에 있다. 이에 본 논문은 후자에 문제의 초점을 맞춰, 관련 문제를 이해, 분석해 보도록 한다.

Keyword: Multicast Protocol, Mbone, Internet, Congestion Control, Fairness Control

1. Introduction

인터넷에 대한 이용 가치의 변화와 다자간 통신 대한 수요 폭발로 말미암아 멀티캐스트 환경(Multicast Environment)이 급속히 발진 중이다. 더불어 기존의 멀티캐스트 프로토콜의 추상적인 일대다 통신 형태로는 복잡한 응용의 요구에 적절히 대응할 수 없는 본질적인 문제에 부딪히면서 응용의 속성을 자세히 분석, 보다 민감하게 적용하는 프로토콜의 연구가 활발히 진행 중이다[1]

한편에서는 실제 인터넷 환경에 1)멀티캐스트 프로토콜을 충돌 없이 전개시키기 위한 연구가 활발히 진행되고 있다. 인터넷의 성공적 발전에 주요 원인중 하나는 네트워크 자원의 적절한 공유에 있다. 공유를 위한 기존의 TCP-기반 혼잡 제어에 독립적인 멀티캐스트 프로토콜의 전개는 네트워크 자원의 독식을 유발시켜 전체 인터넷의 흐름간에 충돌을 빈번히 발생하게 된다. 이러한 흐름간의 충돌을 원만히 해결하기 위해서는 기존의 흐름들과의 유기적 상호 협력 능력을 반드시 갖춰야 하며, 이를 흔히 *TCP-friendly 제어*라 부른다.

더불어, 일대일 TCP-기반 혼잡 제어에서는 고려되지 않았던

다양한 제약 사항들이 멀티캐스트 프로토콜에서는 요구된다. 대표적인 경우가 송신측에서의 *ACK 충돌(ACKnowledgment Implosion)* 현상이다[2]. 윈도우-기반(Window-based)제어 기법은 모든 수신자에게 보내진 패킷에 대한 ACK을 송신측으로 전송하도록 되어 있다. 이를 인식하여, 최종적으로 송신측의 입시영역에 보관되던 이미 보내진 패킷들을 제거한다. 만약, ACK가 일정 시간내에 도착하지 않으면, 입시영역에 보관되어 있던 관련 패킷을 다시 재전송하게 된다.

흔히, 네트워크를 제어하기 위해서 고려되어야 할 사항들은 네트워크의 처리율이 걱정 수치이하로 내려가지 않도록 지원하는 성능 유지 기능, 각 흐름간의 공정성(Fairness), 그리고 성능 유지와 관련된 확장성이라 할 수 있다[3]. 특히, 본 논문에서 두드러지게 고려 될 사항은 기존의 FIFO-기반 라우터상에서 기존의 TCP 흐름과 멀티캐스트 그룹의 흐름간 혹은 멀티캐스트 그룹내의 흐름들간의 공정한 제어를 위한 관리 기법들에 관한 것으로 3장에서 자세히 다루게 된다.

2. Controlled Network

흔히, 혼잡 및 흐름 제어는 이론적, 기능적으로 분류 가능하지만 기능의 성격상 서로 얽혀져 운용되고 있다. 기존의 일대일 TCP기반 네트워크를 제어하기 위한 기능들과 더불어 멀티캐스트 프로토콜의 대두로 말미암아 추가적인 요구 사항들이

1 본 논문에서의 멀티캐스트 프로토콜은 특별한 언급이 없는 한, OSI 모델에서 전송 계층에 해당하며 흔히 멀티캐스트 트랜스포트 프로토콜이라 한다

게시되고 있으며, 이들을 해결하기전에 반드시 고려되어야 할 사항은 기존의 흐름들과 공정한 상호 결합이란 점이다

2.1. Flow Control

대부분의 멀티캐스트 프로토콜들이 명확한 흐름 제어에 대해서 정의해 놓은 경우가 드물다. 흐름 제어와 혼잡 제어는 정의에서 명목적인 구분을 가지고 있으나, 실제 멀티캐스트 환경에서는 서로간의 유기적인 관계가 절대적으로 요구되므로 같이 언급하거나 서로 의존하는 경우가 빈번하다. 일반적으로 흐름 제어는 크게 *Window-based* 흐름 제어와 *Rate-based* 흐름 제어로 나뉘 볼 수 있다[3].

Window-based 흐름 제어에서는 송신측의 일반적인 윈도우에 의지하여 흐름제어를 하는 기법과 수신측에서 각자의 윈도우 크기를 알려 주면, 송신측이 그에 준하여 전송하는 흐름 제어 기법으로 구분된다. 전자의 경우에는 송신측에 모든 ACK의 집중으로 말미암은 ACK 충돌의 발생이며, 후자의 경우에는 각기 다른 윈도우를 가지고 있기 때문에 송신측에서 각각의 윈도우를 어떻게 이해해야 하는 문제점이 있다. 즉, 수신측 하나의 윈도우만을 기준으로 각기 다른 의미를 갖는 수신측의 윈도우 정보를 번의할 수는 없다.

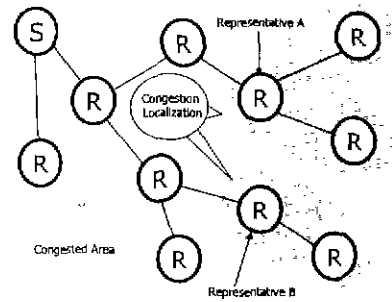
Rate-based 흐름 제어도 마찬가지로 송신측에서 전송율을 측정하는 경우와 각 수신측에서 전송율을 측정하는 경우가 있다. 전송율에 의한 흐름 제어 기법은 특별히 ACK 충돌과 같은 현상은 발생하지 않으나, 전송율을 측정하기 위한 오버헤드가 발생하게 된다.

2.2. Congestion Control

수 많은 흐름들이 네트워크를 공유함에 따라, 특정 네트워크나 네트워크에 많은 흐름이 유입되면 혼잡이 발생하게 된다. 이를 제어할 수 있는 기법을 혼잡 제어라 하며 인터넷이 지금의 모습을 갖출 수 있었던 것도 혼잡 제어 기법의 성공이 한 요인이기도 하다. 따라서, 기존의 흐름들 예를 들어, TCP와 같은 흐름들과 어떻게 적절하게 한정된 네트워크 자원을 공유하는냐는 최근에 발표되는 연구 결과의 양만 보더라도 미루어 짐작할 수 있다.

실제 인터넷상에 멀티캐스트 프로토콜들을 적용하기 위해서는 그룹내의 흐름들간의 혼잡 제어도 이루어져야 하지만 기존의 흐름과의 적절한 타협으로 인하여 유기적 결합 또한 중요한 조건이다. TCP의 혼잡 제어는 윈도우를 기반으로 만약, 혼잡이 발견되면 흐름의 양을 곱으로 줄이고, 다시 혼잡이 없어지면 곱으로 늘리는 이른바 *Fast Recovery - Slow Restart* 기법을 사용하고 있다[4].

트리 구조의 경우 결정적 혼잡의 원인이 지역적 연관 손실과 유사하다는 전제하에 하나의 상위 노드가 혼잡이 발생하게 되면, 하위 계층의 모든 노드들과 같은 혼잡을 경험하게 된다[5]. 트리의 적절한 위치에 작은 대표자 그룹을 설정하여 그들로 하여금, 혼잡 제어와 피드백 충돌들을 해결 하도록 기능을 부여하는 방법이 있다[6]. 송신측은 이들 대표자들과의 통신만으로 전송율을 조절하기 때문에 확장성도 어느 정도 보장한다



[그림 1] 혼잡 국부화(Congestion Localization)

[그림 1]에서 보는 바와 같이 만약 특정 지역에서 혼잡이 발견되면, 그 혼잡 지역의 대표자(A, B)가 동적으로 생성되고, 후에 혼잡이 사라지면 자동으로 대표자도 사라진다. 만약 혼잡이 장기화되게 되면 그 대표자들은 계속해서 모든 일을 대행하게 되는 것이다.

일반적인 데이터의 경우에는 *FEC(Forward Error Correction)* 기법을 이용하여 미리 여러 개로 분할하여 코딩된 다음 특정 주소에 분할된 계층들을 미리 할당해 놓는다[7]. 수신측은 자신의 제약 사항에 맞게 적절히 그룹들에 가입하여 데이터를 가져 오게 된다. 즉, 제약 사항이 완만하면 할수록, 보다 많은 그룹에 가입하여 동시에 가져올 수 있다.

비디오나 오디오 데이터의 경우 많이 쓰이는 기법이 질과 양에 따라 단계적으로 코딩하여 각각의 수신측들이 자신에게 적절한 만큼의 데이터를 수신할 수 있도록 지원하는 *계층적 코딩(Layered Coding)* 기법이 적용된다[8][9]. 즉, 모든 수신측이 같은 제약 사항을 가지고 있는 건 아니기 때문에, 각기 제약 사항에 알맞은 양의 데이터를 수신하기 위해서 미리 질과 양에 따라 코딩된 데이터들을 제공하고 있는 그룹에 적절히 가입함으로써, 혼잡에 적절히 적응하는 방법이다. 만약, 자신이 혼잡을 경험하면 즉시 보다 적은 양의 데이터를 제공하는 그룹에 가입하여 네트워크의 혼잡에 적응해 나아가다 만약 혼잡이 사라지면 다시 보다 높은 질을 제공한 그룹에 가입하는 것이다.

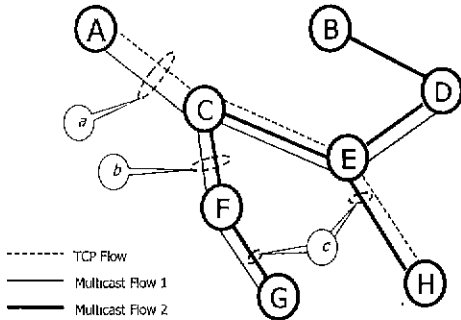
3. Fairness Control

네트워크의 흐름들간에는 흐름에서 다루는 데이터에 따라 흐름의 질이 결정되어야 하지만, 기존의 FIFO 라우터가 주류를 이루는 인터넷상에서는 사실상 불가능하며, 그로 인한 흐름간의 진정한 공정성 제어(*Fairness Control*)는 사실 불가능한 상태다. 공정성 제어는 새로운 연구 분야는 아니다. 기존의 흐름 및 혼잡 제어, QoS 등 흐름과 관련된 연구 분야에서 연관되어서 연구되어야 할 부분이다. 본 논문에서는 멀티캐스트 프로토콜상에서의 공정성과 관련하여 여러 측면에서 분류해 보고, 간단한 알고리즘을 제시해 보도록 한다.

3.1. Intra-Fairness Control

각 멀티캐스트 그룹내에서의 수신자들간의 흐름에 대한 공정성에 대한 제어가 필요하며 이를 *Intra-Fairness Control*이라 하며, 그림[2]의 (c)에 해당한다 예를 들어, 21(E,H)와 1(F,G)는 같은 양의 네트워크 자원이 유용하며, MF(Multicast Flow) 2의 송신자를 D라 가정한다 이 경우 1(C,E)가 혼잡이 발생하게 되면, 1(F,G)는 혼잡의 영향으로 인하여, 1(E,H)와 불공정하게 된다

이러한 불공정을 제어하기 위한 방법으로 두 가지를 제시 할 수 있다 만약 한 지역에서 혼잡이 발생하면 결국 전체 그룹의 성능 저하는 당연한 결과이므로 동적으로 자신의 흐름의 양을 줄여 불필요한 자원의 낭비를 막는 것이다. 이러한 방법을 손실 공정성 제어(Lossy Fairness Control)라 한다. 또 다른 방법으로 혼잡의 발생 여부를 직시하게 되면, 혼잡 지역을 국부화하거나 동적으로 대표자를 설정하여, 그 지역을 비혼잡지역으로부터 은닉시킴으로서, 다른 수신자들에게 피해를 입히지 않는 방법을 들 수 있다. 이를 비손실 공정성 제어(Lossless Fairness Control)라 한다.



[그림 4] 공정성 제어의 설명을 위한 위상

3.2. Inter-Fairness Control

*Inter-Fairness Control*은 기존의 TCP 흐름이나 서로 다른 멀티캐스트 흐름 간의 공정성을 제어하기 위한 것을 말하며, 그림[2]에서 (a), (b)에 속한다 특히, 우리는 그림[2]의 (a)와 TCP와 같은 기존의 유니캐스트 흐름과 멀티캐스트 흐름에 대해서는 *Unicast-to-Multicast Fairness*라 부르며, (b)와 같이 멀티캐스트 그룹끼리 흐름에 대해서는 *Multicast-to-Multicast Fairness*라 부른다 여기서, 해결해야 할 가장 어려운 사실은 기존의 TCP 흐름과 멀티캐스트 프로토콜 흐름과의 공정성 여부를 판단할 수 있는 기준의 설정이다. 사실, 하나의 TCP 흐름은 하나의 수신자만이 관련되지만, 멀티캐스트 프로토콜의 흐름에는 수 많은 수신자들이 연결되어 있다. 즉, 자원 확보의 문제에서 기존의 혼잡이 발생하면 서로 양보하는 방법은 멀티캐스트 프로토콜의 흐름에게 불공정(Unfair)하다는 것이다.

앞서 언급 하였듯이, 공정성 제어는 기존의 혼잡이나 흐름

제어와 밀접한 관련이 있으며, 상호 지원 관계를 반드시 유지해야만 한다. 더불어, 공정성 제어로 인한 오버헤드는 극히 미약해야 하며 동적 흐름의 변화에 민감하게 대응할 수 있어야 하며 결과적으로 확장성을 갖추어야 한다.

4. Work in Progress

현재 본 논문에서 정의된 기본적인 공정성 개념을 바탕으로 다양한 3실험을 전개하고 있다 아직, 어떻게 하면 각 흐름간에 우선 순위를 제공할 수 있는지, 어떤 근거로 인하여 우선 순위가 결정되며, 공정하지 않은 상태의 결정은 어떤 근거로 할 것인지 등 많은 문제들이 산재해 있다.

5. Concluding Remarks

기존 흐름이나 멀티캐스트 그룹내의 흐름들간에 적절하고 공정한 흐름의 제어를 위해서 흐름간에 동등한 비율에 근거한 공정성 제어는 멀티캐스트 프로토콜의 확산과 인터넷의 상용화를 위해 반드시 선결되어야 할 문제이며, 이를 위한 개념들을 조사, 정의해 보았다

Reference

- [1] A. Mankin et al, "IETF Criteria for Evaluating Reliable Multicast Transport and Application Protocols," *Informational RFC 2357, IETF*, Jun 1998
- [2] S. Pingali, D. Towsley and J. F. Kurose, "A Comparison of Sender-Initiated and Receiver-Initiated Reliable Multicast Protocols," *Proc. 1994 ACM SIGMETRICS Conf.*, May 1994
- [3] D.H.H. Guernsey. "Multicast Flow Control in Local Area Networks," *TR95-1479, Cornell University*, 1995.
- [4] W. Stevens., "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms.," *RFC 2001, IETF*, Jan 1997.
- [5] M. Yajnik, J. Kurose, and D. Towsley, "Packet Loss Correlation in the MBone Multicast Network", *IEEE Global Internet Conf.*, Nov 1996.
- [6] D. Delucia and K. Obraczka, "Multicast Feedback Suppression Using Representatives," *Proc. INFOCOMM 97*, Apr 1997.
- [7] L. Rizzo and L. Vicisano, "A Reliable Multicast Data Protocol Based on Software FEC Techniques," *Proc. High Perf. Commun. Sys.*, Jun 1997.
- [8] L. Vicisano, L. Rizzo, and J. Crowcroft, "TCP-like Congestion Control for Layered Multicast Data Transfer," *IEEE INFOCOM 98*, Mar 1998.
- [9] S. McCanne and V. Jacobson, "Receiver-driven Layered Multicast," *ACM SIGCOMM 96*, Aug 1996.

2 1(A,B)는 A노드의 B노드간의 링크를 의미한다.

3 NS(Network Simulator)에서 제공되는 모듈을 이용하고 있다.