

지정된 서버를 이용한 공정한 멀티캐스트 혼잡제어 메커니즘

(A Fair Multicast Congestion Control Mechanism based on the Designated Server)

오재환^{*} 김정현^{**} 정광수^{***} 김화성^{****}
(Jaehwan Oh) (Junghyun Kum) (Kwangsue Chung) (Hwasung Kim)

요약 본 논문에서는 인터넷 망에서 UDP를 사용하는 멀티캐스트 트래픽과 TCP를 사용하는 유니캐스트 트래픽 사이의 공정성 문제를 해결할 뿐만 아니라, 다양한 수신자들의 요구를 공정하게 충족시킬 수 있는 새로운 메커니즘을 제안하였다. 제안한 메커니즘은 송신자 위주의 일률적인 전송률 조정 부담을 지정된 서버로 분산시켰으며, 지정된 서버가 송신자의 데이터를 인터셉트하여 각 서버가 관리하는 로컬 네트워크 상황에 맞게 독립적으로 전송률을 조절하는 방법을 사용하였다. 따라서 제안한 메커니즘은 각 수신자들의 상황을 보다 정확하게 판단하여 서비스를 제공할 뿐만 아니라, 기존의 유니캐스트 세션과의 공정한 네트워크 자원 공유를 실현하였다.

키워드 : 멀티캐스트, 혼잡제어, 공정성

Abstract In this paper, we proposed a new mechanism that solves the fairness problem between unicast traffic using the TCP and multicast traffic using the UDP, and satisfies the requirement of various receivers fairly in the Internet. The proposed mechanism decentralizes the load of blanket transmission rate control from sender to designated server, and uses the method that the designated server intercepts the sender's data and controls the transmission rate suitable for its local network. Therefore, the proposed mechanism not only provides multicast service by accurate estimation of the network status of each receiver, but also realizes both the inter-session fairness and the intra-session fairness problem.

Key words : multicast, congestion control, fairness, RMTP

1. 서론

인터넷이 급속도로 성장하면서 다양한 종류의 서비스를 필요로 하는 어플리케이션들이 등장하였다. FTP나 웹 브라우징, 그리고 전자메일의 경우 송신자와 수신자가 일대일의 통신을 하는 유니캐스트 모델을 사용하지만,

영상회의, 원격강의, 인터넷방송, 컴퓨터 지원 공동 작업(CSCW : Computer Supported Collaborative Work) 등과 같은 어플리케이션은 송신자가 다수의 수신자에게 데이터를 전송하는 방법을 사용한다.

다자간의 통신을 지원하는 하나의 방법으로서는 하나의 송신자가 모든 수신자와 일대일의 유니캐스트 접속을 맺는 방법이 있을 수 있다. 하지만 수신자의 수가 많은 경우 송신자와 네트워크에 있어서 확장성이 문제가 될 수 있다. IP 멀티캐스트는 송신자를 루트로 하고 다양한 목적으로 뻗어나가는 분배 트리로 하나의 데이터만을 송신함으로써 이러한 문제를 해결하였다.

인터넷 멀티캐스트 서비스를 운용하는데 있어서 일대 다수의 멀티캐스트 트랜스포트 프로토콜을 설계하는 것은 혼잡제어라는 근본적인 문제점에 직면하게 되었다.

^{*} 비회원 : 삼성전자 디지털미디어연구소 연구원

jhoh@adams.kwangwoon.ac.kr

^{**} 비회원 : (주)인프로네트워크 연구원

jhgold@inpronetwork.com

^{***} 종신회원 : 광운대학교 전자공학부 교수

kchung@daisy.kwangwoon.ac.kr

^{****} 정회원 : 광운대학교 전자공학부 교수

hskim@daisy.kwangwoon.ac.kr

논문접수 : 2000년 12월 15일

심사완료 : 2002년 3월 11일

네트워크는 많은 사용자들에 의해 공유되고 있으며 따라서 통제되지 않은 데이터 플로우를 멀티캐스트 특성상 네트워크를 상당한 혼잡상황으로 몰고 갈 수 있다. 이러한 문제점을 피하기 위해, 멀티캐스트 프로토콜은 기존의 프로토콜 인스턴스와 네트워크 자원을 공평하게 공유할 수 있도록 설계되어야 한다[1, 2, 3].

멀티캐스트는 비연결형 서비스로써 UDP를 사용하기 때문에 혼잡제어를 하지 않는다. 따라서 TCP와 같이 혼잡제어를 하는 기존의 인터넷 트래픽과 네트워크 자원을 공유하면서 혼잡상황이 발생하였을 경우, 데이터의 손실이 일어나면서 결국 혼잡제어를 하지 않는 멀티캐스트 트래픽만이 네트워크 자원을 독점하는 폐단이 일어난다. 특히 멀티캐스트는 그 특성상 네트워크 전체에 걸쳐 다수의 수신자에게 데이터를 전송하기 때문에 상황에 따라 혼잡상황의 여파가 네트워크 전체에 걸쳐 상당히 크게 영향을 줄 수 있다[4, 5, 6].

기존의 대표적인 신뢰성 있는 멀티캐스트 전송 프로토콜들의 혼잡제어 방식은 모두 송신자가 수신자의 피드백 정보를 통해 일률적으로 데이터 전송률을 조정하는 메커니즘을 사용하고 있다. 이 경우 다양한 수신자들의 네트워크 상태에 부합하지 못하는 일률적인 전송률로 데이터를 전송하는 단점을 갖고 있다.

본 논문에서는 유사한 네트워크 환경을 갖는 각각의 서버 네트워크에 지정된 서버(Designated Server)를 위치킴으로써, 지정된 서버가 송신자로부터 오는 데이터의 전송률을 자신의 서버네트워크의 혼잡상황에 맞게 조정하도록 하는 메커니즘을 제안하였다. 즉, 다양한 네트워크 환경에 산재해 있는 수신자들의 환경에 맞는 공정한 멀티캐스트 혼잡제어가 이루어지도록 하였다. 또한 지정된 서버를 이용하여 세션과 세션간에 공평한 네트워크 자원의 공유가 이루어지도록 하였다.

2. 관계 이론

본 장에서는 멀티캐스트 환경에서의 혼잡제어 개념 및 이를 지원하는 기존의 방식들을 살펴본다.

2.1 멀티캐스트

멀티캐스트란 하나의 송신자가 “한번의 송신으로” 다수의 수신자에게 데이터를 전송하는 것을 말한다. 따라서 일 대 다수의 통신을 하는데 있어서 유니캐스트를 이용할 때와는 달리 송신자 입장에서는 같은 데이터를 다수의 수신자들에게 반복해서 전송할 필요가 없으므로 송신자의 작업 부하가 감소되며, 같은 데이터가 네트워크상의 경로를 중복해서 지나는 경우를 최소화함으로써

자원을 보다 효율적으로 이용할 수 있는 것이다[7].

멀티캐스트를 하기 위해서는 먼저 라우터가 이를 지원해줘야 하는데, 현재 인터넷상에 존재하는 대부분의 라우터가 이를 지원하지 못한다. 따라서 멀티캐스트를 지원하는 멀티캐스트 라우터(MRouter)가 이를 지원하지 못하는 유니캐스트 라우터를 경유해 다른 멀티캐스트 라우터에게 데이터를 전송하기 위한 방법으로 멀티캐스트 라우터간에 ‘터널링(Tunneling)’이라는 연결을 설정함으로써 가상의 네트워크를 인터넷상에 구성하고 있으며 이를 MBone(Multicast Backbone)이라 한다[8, 9]. 그림 1은 터널링에 대한 개념도를 나타내었다.

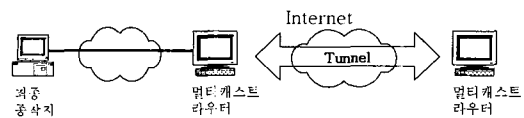


그림 1 터널링

2.2 멀티캐스트 환경에서의 혼잡제어

멀티캐스트의 효율성이 부각되면서 다양한 알고리즘들이 제안되었고 그에 따른 여러 가지 문제점들도 제기되었다. 특히 기존에 효율적인 트리 형성의 연구관점에서 벗어나, 최근에는 확장성 문제와 더불어 멀티캐스트 환경에서의 혼잡제어 방식에 관한 연구가 현재 IRTF 산하 RMRG에서 활발히 진행되고 있다.

기존의 유니캐스트 트래픽으로써 TCP의 경우 네트워크에 혼잡상황이 발생되면 그에 따른 적절한 혼잡제어 메커니즘을 수행함으로써 가용자원을 다른 세션과 공평히 사용할 수 있는 방안을 제공하지만 UDP의 경우 네트워크 상황에 따른 혼잡제어 메커니즘을 지원하지 않는다. 따라서 TCP 트래픽과 UDP 트래픽이 공유 자원을 사용하면서 혼잡상황이 발생하여 패킷이 손실될 경우, TCP만이 데이터의 전송률을 낮추면서 혼잡제어를 하기 때문에 결과적으로 UDP 트래픽만이 네트워크 공유자원을 대부분 사용하게 된다. 멀티캐스트는 기본적으로 비연결형 서비스로써 UDP를 사용하기 때문에 혼잡제어 메커니즘을 제공하지 않는다. 따라서 TCP 세션과 멀티캐스트 세션이 부족한 네트워크 자원을 공유할 경우 멀티캐스트 세션이 대부분 독점하게 되는 것이다[10].

멀티캐스트 환경에서의 혼잡제어는 공정성(fairness)을 그 관점으로 잡을 수 있다. 공정한 멀티캐스트 혼잡제어라는 것은 두 가지 접근 방식으로 나눌 수가 있는데, 세션과 세션과의 공정성 문제(inter-session fairness)와 세션 내부에서의 공정성 문제(intra-session fairness)가

그것이다. 세션과 세션간의 공정성 문제라는 것은, 같은 종류의 세션 즉, 하나의 멀티캐스트 세션과 또 다른 멀티캐스트 세션간에 네트워크 자원을 얼마만큼 공평하게 나누어 사용하는가를 살펴보는 것과, 다른 종류의 세션 즉, 멀티캐스트 세션과 유니캐스트 세션간에 네트워크 자원을 얼마만큼 공평하게 나누어 사용하는가를 살펴봄으로써 판단할 수 있다.

세션 내부에서의 공정성 문제는 하나의 송신자를 기준으로 볼 때 다수의 수신자가 자신의 네트워크 상황에 맞게 데이터를 수신할 수 있는지의 가용 여부를 놓고 판단한다. 즉, 멀티캐스트 특성상 다수의 수신자가 하나의 송신자로부터 일률적으로 조정된 데이터 전송률로 수신을 하기 때문에, 송신자가 네트워크 자원이 풍부한 수신자를 기준으로 데이터를 전송하게 되면 네트워크 가용 자원이 부족한 수신자는 많은 양의 데이터를 모두 수신하지 못하고 폐기하게 된다. 반대로 송신자가 네트워크 자원이 부족한 수신자를 기준으로 데이터를 전송하게 되면 네트워크 자원이 풍부한 수신자는 자신의 네트워크 가용자원에 비해 부족한 데이터 전송률로 수신하는 폐단을 받게된다. 멀티캐스트 환경에서 혼잡제어라는 것은 이러한 세션과 세션과의 공정성 문제와 세션 내부에서의 공정성 문제를 다룸으로써, 멀티캐스트 트래픽과 기존의 인터넷 트래픽과의 공정성을 부여하고, 멀티캐스트 특성상 일어날 수 있는 수신자들간의 불공정성 문제를 해결하는 것이다.

2.3 기존 메커니즘

멀티캐스트 환경에서의 혼잡제어 방식은 신뢰성 있는 멀티캐스트 전송 프로토콜을 개발하는 과정에서 제안되었다. SRM(Scalable Reliable Multicast)이나 ARM(Active Reliable Multicast)과 같은 대표적인 멀티캐스트 전송 프로토콜이 명시적인 혼잡제어 메커니즘이 없는 것처럼 최근에 와서야 멀티캐스트 혼잡제어 방식에 대해 많은 메커니즘들이 제안되었다. 이러한 메커니즘들은 하나의 공통점을 가지고 있는데, 멀티캐스트 송신자의 데이터 전송률을 조절함으로써 다수의 세션들간에 네트워크 자원(bandwidth)을 공평하게 이용하자는 것이다. 표 1에 여러 가지 신뢰성 있는 멀티캐스트 전송 프로토콜에서 사용하는 혼잡제어 방식을 나열하였다[11, 12].

표 1 멀티캐스트 프로토콜의 혼잡제어 메커니즘

	PGM	ARM	RMTP	SRM	AER/NCA
공정 제어	수신자의 보류 NACK에 대해 혼잡제어	명시되지 않음	태그 수신제어 및 첫 수신자에 대해 송신자가 혼잡제어	명시되지 않음	태그 수신제어 및 첫 수신자에 대해 송신자가 혼잡제어

2.3.1 PGM(Pragmatic General Multicast)

PGM은 1998년 1월 Cisco 시스템에 의해 처음 문서화되었고 그해 2월에 IRTF 산하 RMRG 회의에서 처음 제안되어졌다. 처음 PGM이 제안되었을 당시 혼잡제어에 대한 메커니즘은 포함되어 있지 않았고, 그 후 새로운 버전에 추가되었다[13].

PGM은 수신자들이 데이터의 수신여부를 판단하기 위해 NACK 신호를 이용한다. 즉, 모든 수신자들이 데이터를 전송 받으면서 손실된 패킷에 대해서만 NACK 신호를 송신자에게 전송함으로써 데이터의 손실상태를 알려주게 된다. 송신자는 그룹 내의 모든 수신자들로부터 전송되어 오는 NACK 신호를 기초로 해서 데이터의 전송률을 조정하는 혼잡제어 방식을 사용한다.

2.3.2 RMTP(Reliable Multicast Transport Protocol)

RMTP는 AT&T산하에 있었던 당시 Bell 연구소에 의해 처음 개발되어졌다. RMTP의 혼잡제어 메커니즘은 네트워크의 계층성에 의지한다. 즉, RMTP는 네트워크를 계층적으로 구성하고 각각의 계층에 지정된 수신자(Designated Receiver) 선정하여 수신자들의 피드백을 하나로 수렴한다[14]. 이렇게 하나로 수렴된 피드백이 송신자에게 전송되고 송신자는 피드백에 기초하여 데이터의 전송률을 조정하게 된다. RMTP는 PGM과는 달리 피드백 신호로써 ACK 신호를 이용하며, 각 계층에 있는 지정된 수신자가 같은 계층에 있는 나머지 수신자들의 피드백을 하나로 수렴하여 송신자에게 전송하게 된다. 그림 2는 RMTP에서 지정된 수신자로 ACK가 하나로써 수렴되는 과정을 보여준다.

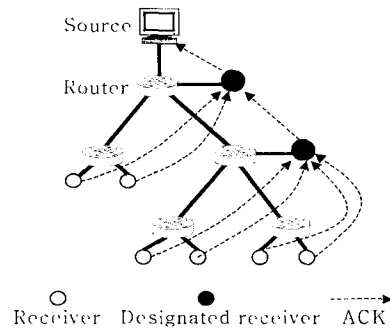


그림 2 RMTP 메커니즘

2.3.3 AER/NCA(Active Error Recovery/Nominee Congestion Avoidance)

AER/NCA는 Bell 연구소의 Kascra에 의해 제안되었다. AER/NCA의 혼잡제어 메커니즘은 RMTP와 상

당히 유사하다. 즉, 네트워크를 계층적으로 구성하고, 각 계층에 하나의 복구 서버(Repair Server)를 위치시킴으로써 각각의 복구 서버가 수신자들이 피드백을 수렴한다. 각 계층에서 하나로 수렴된 수신자들의 피드백은 상위 계층의 복구 서버에게 전송이 되고 최종적으로 송신자에게 하나의 피드백만을 전송하게 된다. 송신자는 하나로 수렴되어 전송되어 온 피드백에 따라 데이터의 전송률을 조정하게 된다[15]. AER/NCA는 피드백 신호를 NACK 신호를 이용한다는 점이 RMTP와 다르다.

3. 지정된 서버를 이용한 멀티캐스트 혼잡제어 메커니즘

본 장에서는 멀티캐스트 환경에서 혼잡을 제어하는 기존 메커니즘들의 문제점을 분석한 후, 이러한 문제점을 해결하기 위하여 본 논문에서 제안한 메커니즘에 대하여 설명한다.

3.1 기존 메커니즘의 문제점

Speakman이 제안한 PGM은 송신자가 데이터 전송률을 조정하기 위해 모든 수신자들로부터 전송되는 NACK 신호를 이용했다. 즉, 송신자는 모든 수신자들에게 같은 전송률로 데이터를 전송하게 되고, 수신자들은 자신들의 네트워크 상황에 맞게 데이터를 수신하면서, 수신하지 못한 패킷의 NACK 신호를 송신자에게 전송하게 된다. 이러한 경우 수신자들의 수가 많을 때 송신자에게서 NACK 신호의 폭주 현상이 일어날 수 있으며, 송신자가 모든 NACK 신호를 처리해야 하는 부담이 생겨 확장성에 어려움이 있다. RMTP와 AER/NCA는 수신자들 중에 가장 낮은 throughput을 갖는 수신자를 선정하여 그에 대한 피드백을 통해 송신자가 데이터의 전송률을 조정한다. 즉, RMTP는 네트워크를 계층적으로 구성하고 각 계층의 지정된 수신자(Designated Receiver)를 이용하여 서브트리상의 수신자들에 대한 데이터 피드백을 하나로 수렴함으로써 폭주현상을 방지하였고, 하나로 수렴된 수신자의 피드백에 따라 송신자의 데이터 전송률을 조정하였다.

AER/NCA의 경우도 RMTP와 마찬가지로 계층적으로 구성된 네트워크상의 복구 서버(Repair Server)를 이용하여 서브트리상의 수신자들에 대한 데이터 피드백을 하나로 수렴하여 피드백 폭주를 방지하였고, 하나의 대표 수신자(nominee)를 선정하여 그에 대한 피드백을 이용하여 송신자의 데이터 전송률을 조정하였다. 즉, RMTP와 AER/NCA는 송신자가 모든 수신자들에 대하여 일률적으로 같은 전송률을 적용한다.

앞에서 볼 수 있듯이 기존의 멀티캐스트 혼잡제어 방식

은 모두 하나의 공통점을 가지고 있다. 즉, 다양한 수신자들의 네트워크 환경을 고려하지 않고 송신자 위주의 일률적인 전송률을 모든 수신자들에게 적용함으로써 수신자들의 다양한 혼잡상황에 대한 수용 능력이 부족하다. 예를 들면 그림 3에서 R4에 생긴 병목구간으로 인하여 송신자가 수신자 R4의 가용자원에 맞는 0.2Mbps의 전송률로 데이터를 전송하게 된다면 R1, R2, R3의 나머지 수신자들은 1Mbps의 충분한 대역폭이 있음에도 불구하고 0.2Mbps의 낮은 전송률로 데이터를 수신하는 불공정성을 초래하게 된다. 반대로 송신자가 대역폭이 충분한 수신자 R1, R2, R3의 가용자원에 맞추어 1Mbps의 전송률로 데이터를 전송하게 된다면, 대역폭이 충분하지 못한 R4의 수신자는 자신의 수용할 수 있는 능력보다 빠른 전송률로 오는 모든 데이터를 수신하지 못하고 많은 패킷을 폐기하게 되는 것이다.

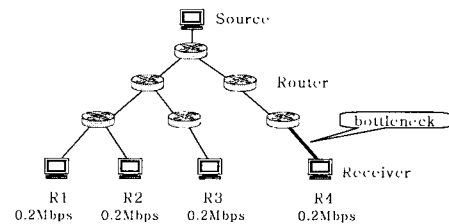


그림 3 수신자들의 불공정성

3.2 제안한 메커니즘

본 논문에서는 앞에서 제기된 기존의 혼잡 제어 메커니즘에 대한 문제점을 해결하기 위해 지정된 서버(Designated Server)를 이용한 공정한 멀티캐스트 혼잡제어 메커니즘을 제안하였다. 즉, 다양한 수신자들의 네트워크 환경을 고려하지 않고 송신자 위주의 일률적인 전송률을 모든 수신자들에게 적용한 기존의 메커니즘을 보완하여, 지정된 서버에서 자신이 관리하는 서브네트워크의 혼잡 상황에 맞게 전송률을 조정하도록 하였다. 예를 들면, 송신자로부터 전송되는 데이터를 그림 4와 같

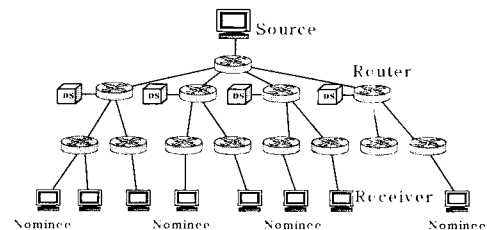


그림 4 제안한 메커니즘

이 각 서브네트워크에 위치하고 있는 지정된 서버가 인터셉트 한 후 자신의 서브네트워크의 혼잡상황에 맞게 데이터 전송률을 조정함으로써 수신자들의 혼잡제어 부담을 각각의 지정된 서버로 분산시켰으며, 다양한 수신자들의 혼잡 상황을 수용함으로써 세션 내부에서의 수신자들의 불공정성 문제를 해결하였다.

3.2.1 대표 수신자(nominee) 선정

멀티캐스트 환경에서 혼잡제어를 하기 위해 가장 주의해야 할 점은 유니캐스트 세션 혹은 또 다른 멀티캐스트 세션과의 공정성이다. 이러한 목적을 달성하기 위해 각각의 서브네트워크에 있는 지정된 서버는 자신의 서브네트워크에 하나의 대표 수신자를 선정하고 이러한 대표 수신자로부터의 패킷 손실 상태에 따라 전송률을 조정하게 된다. 유니캐스트 세션과의 공정성을 얻기 위해 TCP와 유사한 방식으로 혼잡제어 메커니즘이 설계되었다.

우선, 고정크기의 패킷 사이즈를 갖는 TCP 세션의 평균 대역폭은 패킷 손실 확률($p = \text{손실된 패킷 수} / \text{전체 패킷 수}$) 과 RTT(패킷의 타임 스템프 정보를 이용하여 계산)를 인자로 갖는 다음과 같은 식으로 나타낼 수 있다.

$$B(p, T) = C / (T * \sqrt{p}) \quad (1)$$

여기서, (1)번식의 우변의 분모를 g 함수로 정의하면,

$$g(p, T) = T * \sqrt{p} \quad (2)$$

가 된다. (2)번식에서, 함수 g 의 값이 커질 때 대역폭은 낮아지므로 g 함수의 값이 가장 큰 경로상의 맨 종단에 위치한 수신자가 대표 수신자로 선정된다. 대표 수신자는 주기적으로 자신의 서브네트워크에 있는 지정된 서버에게 패킷 손실 확률과 RTT(Round Trip Time)값이 포함된 혼잡 상태 메시지를 전송하며, 지정된 서버는 수신자들로부터 전송 받은 혼잡 상태 메시지를 이용하여 g 함수 값을 계산하게 된다. 대표 수신자의 선정 과정을 정리하면 다음과 같다.

- 각각의 수신자는 멀티캐스트 세션에서 송신자로부터 전송되어 오는 데이터의 손실률에 따라 수신자까지의 패킷 손실 확률 값(p)과 RTT(T)값을 계산한다. 이러한 수신자들은 패킷 손실 확률 값(p)과 RTT(T)값을 혼잡상태 메시지(Congestion Status Message: CSM)에 담아 제어 정보로써 주기적으로 자신의 지정된 서버에게 전송한다.

- 자신의 서브네트워크에 있는 수신자들로부터 주기적으로 수신된 혼잡 상태 메시지를 이용하여, 지정된 서버는 식(2)를 기반으로 해서 g 함수의 값이 가장 큰 경로상의 맨 끝에 위치한 수신자를 대표 수신자로 선정한다.

- 대표 수신자가 선정되면, 지정된 서버는 자신의 서브네트워크에 있는 대표 수신자에게 패킷 당 ACK를 전송하도록 요구한다.

3.2.2 전송률 조정 알고리즘

지정된 서버는 송신자로부터 전송되는 데이터를 인터셉트하면서 자신이 관리하는 대표 수신자와의 피드백 정보(패킷 당 ACK)를 통해 전송률을 조정한다. 지정된 서버의 전송률 조정 알고리즘은 다음과 같다.

- ACK를 받았을 때, Window의 크기 $W(\text{packet})$ 가 임계 값보다 작으면 1을 증가시키고, 그렇지 않으면 $1/W$ 만큼 증가시킨다.

- 데이터의 손실을 발견했을 때, 임계 값을 현재 Window 크기의 반으로 줄이고, Window의 크기도 반으로 줄인다.

- 시간이 초과되었을 때, 임계 값을 현재 Window 크기의 반으로 줄이고, Window의 크기도 1로 줄인다.

패킷이 전송된 후 다음 세 개의 패킷에 대한 ACK를 받는 동안 처음 패킷에 대한 ACK를 받지 못하면 그 패킷은 손실된 것으로 가정한다. 또한 가장 최근에 패킷을 전송한 후 일정시간 동안 어떠한 ACK도 받지 못하면 타임아웃으로 간주한다.

3.2.3 라우터의 지원

지정된 서버를 이용한 전송률 조정을 위해서는 라우터가 지원해야 하는 두 가지 기능이 있다. 두 가지 새로운 기능은, 송신자로부터 전송되어 오는 데이터를 라우터가 인터셉트해 지정된 서버로 보내주는 기능과 다시 지정된 서버로부터 혼잡상황에 맞게 전송률이 조정되어 오는 데이터를 자신의 서브네트워크에 서브캐스팅 해주는 기능이다.

데이터의 인터셉트는 IP 헤더의 옵션 필드를 이용한다. 이러한 옵션을 IP 라우터 경고 옵션(IP router alert option)이라 정의하고 지정된 서버와 같은 레벨로 배치되어 있는 라우터(collocated router)가 인식하여 그림 5와 같이 지정된 서버로 보내주게 된다.

데이터의 서브캐스팅은 IP encapsulation을 이용하여

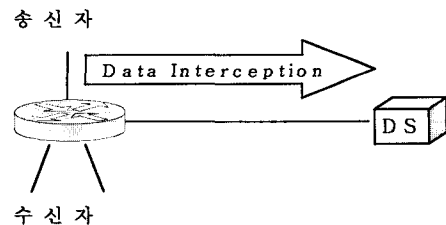


그림 5 데이터의 인터셉션

다음과 같이 얻을 수 있다. 우선 서브캐스트 패킷은 지정된 서버와 같은 레벨로 배치되어 있는 라우터(collocated router)로 유니캐스트 된다. 이러한 유니캐스트 패킷은 원래의 멀티캐스트 송신자 주소를 송신자 주소로 하고 멀티캐스트 그룹 주소를 목적지 주소로 하는 IP 패킷 헤더를 포함하고 있다. 지정된 서버로부터 유니캐스트된 패킷을 받으면, 라우터는 유니캐스트 패킷을 decapsulation한 후 자신의 서브트리상으로 원래의 송신자로부터 전송되는 것처럼 그림 6과 같이 멀티캐스트 하게 된다. 이렇게 함으로써 데이터가 원래의 송신자에게 재전송 되는 것을 방지할 수 있다.

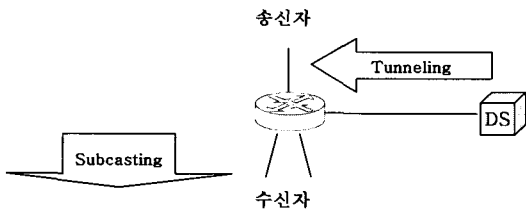


그림 6 터널링과 서브캐스팅

4. 실험 및 결과

본 장에서는 제한한 알고리즘에 대한 세션과 세션과의 공정성 및 세션 내부에서의 공정성을 측정하기 위해 시뮬레이션을 통하여 검증하였다. 공정성을 검증하기 위한 시뮬레이션 툴(tool)로써는 LBNL(Lawrence Berkeley National Laboratory)의 ns2(network simulator 2)를 이용하였으며[16], 모든 링크는 1Mbps의 대역폭과 10ms의 전송지연 시간을 가지며 라우터는 drop-tail 방식을 사용하였다. 운용 OS(Operating System)은 Linux 6.1 Kernel 2.2.12 버전을 사용하였다. 시뮬레이션은 세션과 세션간의 공정성과 세션 내부에서의 공정성을 측정하기 위해 네 단계로 구분하여 테스트하였다. 처음 세 단계까지의 실험은 서로 다른 종류의 세션과 같은 종류의 세션, 다양한 종류의 세션을 선택하여 시뮬레이션 함으로써 세션간의 공정성(inter-session fairness)을 측정하였고, 마지막 네 번째 단계의 실험은 세션 내부에서의 공정성(intra-session fairness)을 측정하였다.

4.1 지정된 서버를 이용한 서로 다른 세션간의 공정성 측정

시뮬레이션 네트워크를 그림 7과 같이 구성하고 유니캐스트 세션과 멀티캐스트 세션이 하나의 병목 구간을

공유할 때 두 가지 서로 다른 종류의 세션, 즉 멀티캐스트 세션과 유니캐스트 세션이 같은 병목 구간의 대역폭을 얼마만큼 공정하게 공유하는지에 대한 공정성을 측정하였다. 시뮬레이션 결과 멀티캐스트 세션은 1Mbps의 링크 대역폭 중에 0.45Mbps의 대역폭을 점유하였으며 유니캐스트 세션은 0.44Mbps의 대역폭을 점유하였다. 즉, 링크 대역폭을 서로 공정하게 공유했음을 확인할 수 있었으며 그 결과를 그림 8에 나타내었다.

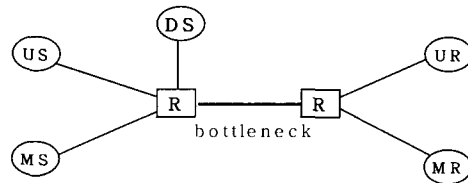


그림 7 시뮬레이션 구성 A

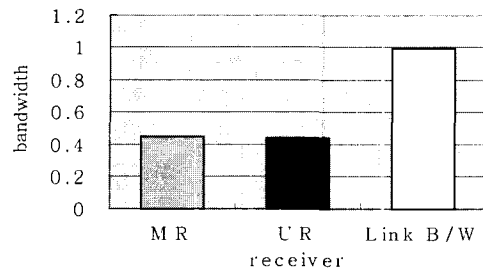


그림 8 구성 A의 시뮬레이션 결과

4.2 지정된 서버를 이용한 같은 종류의 세션간의 공정성 측정

두 번째 시뮬레이션은 같은 종류의 멀티캐스트 세션이 하나의 병목 구간을 공유할 때 세션간의 대역폭 공유 공정성을 측정하였다. 시뮬레이션 네트워크는 그림 9와 같이 구성하였으며, 시뮬레이션 결과 멀티캐스트 수신자 MR1은 0.47Mbps의 대역폭을 점유하였으며 MR2는 0.45Mbps의 대역폭을 점유하였다. 즉, 같은 종류의

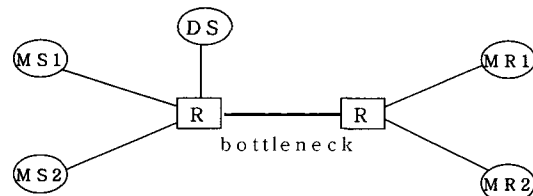


그림 9 시뮬레이션 구성 B

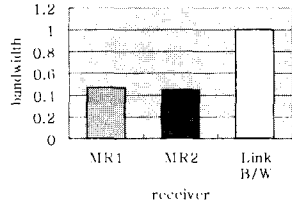


그림 10 구성 B의 시뮬레이션 결과

멀티캐스트 세션간에 공정한 대역폭 공유가 이루어졌음을 확인하였다. 결과는 그림 10에 나타내었다.

4.3 지정된 서버를 이용한 다양한 세션간의 공정성 측정

세 번째 시뮬레이션은 좀더 다양한 세션들이 병목구간을 얼마만큼 공정하게 공유하는가에 대한 측정을 위해 그림 11과 같이 네트워크를 구성하고 테스트하였다. 이러한 네트워크의 경우 두 개의 병목구간이 발생하고, 유니캐스트 트래픽과 멀티캐스트 트래픽 양쪽의 병목구간으로 똑같이 집중되도록 설정하였다.

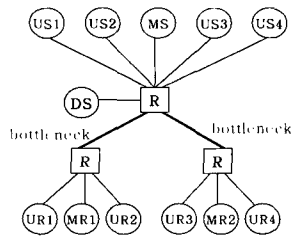


그림 11 시뮬레이션 구성 C

시뮬레이션 결과 멀티캐스트 세션의 수신자 MR1과 MR2는 모두 0.22Mbps 대역폭을 점유하였으며, 유니캐스트 세션의 수신자 UR1, UR2, UR3, UR4는 각각 0.23Mbps, 0.27Mbps, 0.20Mbps, 0.27Mbps의 대역폭을 점유하였음을 확인할 수 있었다. 즉, 멀티캐스트 세션의 수신들과 유니캐스트 세션의 수신자들 사이에 공정하게 대역폭 공유가 이루어짐을 확인할 수 있었고 그 결과물 그림 12에 나타내었다.

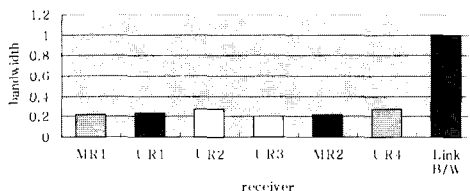


그림 12 구성 C의 시뮬레이션 결과

4.4 지정된 서버를 이용한 세션 내부에서의 공정성 측정

네 번째 시뮬레이션은 두 개의 멀티캐스트 수신자중 하나(MR2)가 다른 유니캐스트 세션과 하나의 병목구간을 공유할 때 유니캐스트 세션과 얼마만큼 공정하게 대역폭을 공유하는가에 대한 측정과, 또 하나의 멀티캐스트 수신자(MR1)가 다른 멀티캐스트 수신자(MR2)의 가용자원과 상관없이 자신의 가용자원에 맞게 데이터를 수신할 수 있는가에 대한 측정을 하였다. 시뮬레이션 네트워크는 그림 13과 같다.

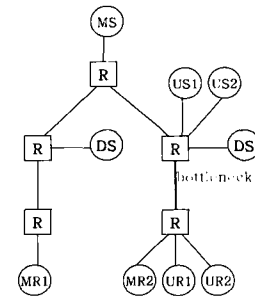


그림 13 시뮬레이션 구성 D

시뮬레이션 결과 멀티캐스트 수신자 MR2는 0.27Mbps의 대역폭을 점유하였고, MR2와 하나의 병목구간을 공유하고 있는 다른 유니캐스트 세션의 수신자 UR1, UR2는 각각 0.22Mbps와 0.21Mbps의 대역폭을 점유하였다. 즉, MR2와 UR1, UR2가 공정하게 대역폭을 점유함을 확인할 수 있었고, 또 다른 멀티캐스트 수신자 MR1은 0.86Mbps의 대역폭을 점유함으로써 MR2의 데이터 전송률과 상관없이 자신의 네트워크 가용자원에 맞게 빠른 데이터 전송률로 수신할 수 있었음을 확인하였다. 그 결과물 그림 14에 나타내었다.

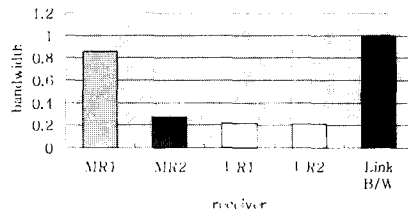


그림 14 구성 D의 시뮬레이션 결과

5. 결론 및 향후과제

현재 많은 연구가 진행되고 있는 멀티캐스트는 일 대

다수의 통신을 하는데 있어서 효율적인 데이터 전송 수단을 제공하지만, 혼잡제어를 하지 않는 UDP를 이용하기 때문에 기존의 TCP를 사용하는 유니캐스트 트래픽과의 공정성 문제를 야기 시킨다. 이러한 문제를 해결하기 위한 기존의 멀티캐스트 혼잡제어에 관한 연구는, 송신자가 일률적으로 전송률을 조정하는 방식을 사용하기 때문에 송신자의 부하가 클 뿐만 아니라 수신자들의 다양한 네트워크 환경을 고려하지 못하는 문제점이 있다.

본 논문에서는 멀티캐스트 트래픽과 TCP를 사용하는 유니캐스트 트래픽과의 공정성 문제를 해결하면서 수신자들의 다양한 네트워크 환경을 고려하는 새로운 메커니즘을 제안하였다. 제안한 메커니즘은 지정된 서버를 이용하여 혼잡 발생 시에 전송률을 조정함으로써 공정성 문제를 해결하였으며, 지정된 서버는 각 서브네트워크에 있는 대표수신자의 피드백 정보를 통해 수신자들의 다양한 네트워크 환경 맞도록 데이터 전송률을 조정하였다. 즉, 각각의 서브네트워크에 있는 지정된 서버가 송신자로부터 전송되어 오는 데이터를 인터셉트하여 자신의 네트워크 환경에 맞게 전송률을 조정하였으며, 또한 혼잡제어 부담을 여러 개의 서버로 분산시켜 송신자의 부담을 감소시켰다. 시뮬레이션을 통하여 제안한 메커니즘을 검증한 결과 각각의 지정된 서버를 갖는 서브네트워크에서 서로 다른 세션들, 즉 유니캐스트 세션과 멀티캐스트 세션 혹은 멀티캐스트 세션과 또 다른 멀티캐스트 세션간에 공정한 대역폭 공유를 확인하였고 수신자들의 다양한 네트워크 환경에 맞게 전송률을 조정함을 확인하였다.

향후 연구과제로는 지정된 서버를 계층적으로 구성해서 각 계층에 있는 수신자들의 혼잡 상태 메시지를 하나로 수렴하고, 각 계층 레벨의 지정된 서버에서 혼잡제어를 수행함으로써 좀더 지역적인 혼잡제어를 하는 것이다.

참 고 문 헌

- [1] J. Padhye et al., "A TCP-Friendly Rate Adjustment Protocol for Continuous Media Flows Over Best Effort Network," Proc. NOSSDAV '99, June 1999.
- [2] I. Rhee, N. Ballaguru, and G. N. Rouskas, "MTCP: Scalable TCP-like Congestion Control for Reliable Multicast," Proc. IEEE INFOCOM '99, March 1999.
- [3] L. Vicisano, L. Rizzo, and J. Crowcroft, "TCP-Like Congestion Control for Layered Multicast Data Transfer." Proc. IEEE INFOCOM '98, March 1998.
- [4] J. Golestani and K. Sabnani, "Fundamental Observations on Multicast Congestion Control in the Internet," Proc. IEEE INFOCOM, March 1999.
- [5] S. J. Golestani and S. Bhattacharyya, "A Class of End-to-End Congestion Control Algorithms for the Internet," Proc. IEEE ICNP '98, Oct. 1998.
- [6] D. Sisalem and H. Schulzrinne, "The Loss-Delay Adjustment Algorithm: A TCP-Friendly Adaptation Scheme," Proc. NOSSDAV '98, July 1998.
- [7] S. E. Deering and D. R. Cheriton, "Multicast Routing in Datagram Internetworks and Extended LANs," ACM Trans. on Computer Systems, vol. 8, no. 2, pp. 85-110, May 1990.
- [8] S. E. Deering, et. al, "An Architecture for Wide Area Multicast Routing," ACM SIGCOMM '94, pp. 126-135, Aug. 1994.
- [9] V. P. Kompella, J. C. Pasquale, and G. C. Polyzos, "Multicast Routing for Multimedia Communications," IEEE/ACM Trans. on Networking, vol. 1, no. 3, pp. 286-292, June 1993.
- [10] J. Mahdavi and S. Floyd, "TCP-Friendly Unicast Rate-Based Flow Control," Note in the end2end-internet mailing list, Jan. 1997.
- [11] L. Lehman, S. Garland, and D. Tennenhouse, "Active Reliable Multicast," Proc. IEEE INFOCOM, March 1998.
- [12] S. Kasera, J. Kurose, and D. Towsley, "A Comparison of Server-Based and Receiver-Based Local Recovery Approaches for Scalable Reliable Multicast," UMass tech. rep., Oct. 1998.
- [13] T. Speakman et al., "Pragmatic General Multicast," Internet Draft, Aug. 1998.
- [14] S. Paul et al., "RMTP: A Reliable Multicast Transport Protocol," IEEE JSAC, vol. 15, no. 3, pp. 407-21, April 1997.
- [15] Sneha K. Kasera and Supratik Bhattacharyya, "Scalable Fair Reliable Multicast Using Active Services," Proc. IEEE NETWORK, Jan. 2000.
- [16] "UCB/LBNL/VINT network simulator ns (version 2)," <http://www-mash.cs.berkeley.edu/ns/>



오 재 환

1999년 광운대학교 전자통신공학과 학사.
2001년 광운대학교 전자통신공학과 석사.
2001년 ~ 현재 삼성전자 디지털 미디어 연구소 연구원. 관심분야는 컴퓨터통신, 멀티미디어통신



금 정 현

1997년 광운대학교 전자통신공학과 학사.
1999년 광운대학교 전자통신공학과 석사.
2001년 8월 광운대학교 전자통신공학과 박사 수료. 현재 (주)인프로네트워크 선임연구원. 관심분야는 라우팅 프로토콜, 실시간 인터넷 프로토콜, QoS



정 광 수

1981년 한양대학교 전자공학과 학사.
1983년 한국과학기술원 전기 및 전자공학과 석사. 1991년 미국 University of Florida 전기공학과 박사(컴퓨터공학전공). 1983년 ~ 1993년 한국전자통신연구원 선임연구원. 1991년 ~ 1992년 한국과학기술원 대우교수. 1993년 ~ 현재 광운대학교 전자공학부 부교수(신기술연구소 연구원). 관심분야는 멀티미디어통신, 컴퓨터통신, 분산처리



김 화 성

1981년 2월 고려대학교 전자 공학과 졸업. 1983년 2월 고려대학교 전자 공학과 석사. 1996년 10월 Lehigh univ. 전산학 박사. 1984년 3월 ~ 2000년 2월 ETRI 책임 연구원. 2000년 3월 ~ 현재 광운대학교 전자공학부 교수. 관심분야는 차세대 인터넷 구조, 미들웨어 환경, 그리드 컴퓨팅