

QoS와 신뢰성을 제공하는 확장성 있는 오버레이 멀티캐스트

이 춘 성[†] · 한 선 영^{**} · 송 정 욱[†] · 최 병 욱[†]

요 약

본 논문에서는 확장성 있고 신뢰성 있는 오버레이 멀티캐스트를 지원 하기 위하여 새로운 멀티캐스트 트리 구성 방법과 지역적인 멀티캐스트를 도입하였다. 본 논문에서 제안하는 구조는 멀티캐스트 네트워크와 멀티캐스트를 지원하지 않는 네트워크가 혼합된 환경에서 멀티캐스트 소스로부터 사용자들에게 스트리밍 데이터를 효율적으로 전달할 수 있다. 새로운 멀티캐스트 트리 구성 방식은 서브넷에 릴레이가 존재하지 않는 경우에 최적의 조건인 원격 릴레이와 접속을 가능하게 한다. 또한, IPv6 헤더의 Traffic Class에 새로운 서비스 타입을 정의하여 릴레이에서 선택적인 FEC 적용과 차별화된 전송을 가능하게 하였다. 우리가 제시한 해결책의 효과를 검증하기 위하여, 실제 서비스 환경과 유사한 환경을 시뮬레이터로 구현하고 테스트 하였으며, 그 테스트 결과는 전통적인 유니캐스트 접근 방식과 기존의 오버레이 멀티캐스트 방식보다 훨씬 더 효율적인 것으로 판명되었다.

키워드 : 오버레이 멀티캐스트, 확장성, 신뢰성, IP 버전 6, 차별화된 서비스

Scalable Overlay Multicast supporting QoS and Reliability

Choonsung Rhee[†] · Sunyoung Han^{**} · Jungwook Song[†] · Byounguk Choi[†]

ABSTRACT

In order to support overlay multicast with scalability and reliability, in this paper, we introduced a new multicast tree construction method and a regional multicast. The architecture we introduce efficiently transmits streaming data to the users from multicast source in an environment in which non-multicast network and multicast network. The new multicast tree construction method enables the connection with a remote relay which is an optimum condition in case there is no relay in subnet. Besides, by defining a new service type to traffic class of IPv6 header, a selective adoption of FEC and a distinctive transmission became possible. In order to verify the effect of the solution we suggested, we embodied an environment which is similar to the actual service environment by a simulator and tested it. The result of the test shows that the proposed method is more efficient than the traditional unicast approach method and the existing overlay multicast.

Key Words : Overlay Multicast, Scalability, Reliability, IPv6 and QoS

1. 서 론

IP 멀티캐스트 도입의 부진은 현재 네트워크 기반을 변경하지 않은 채 어플리케이션 레벨의 상위 계층에서 멀티캐스트의 기능을 처리할 수 있는 오버레이 멀티캐스트를 등장하게 하였다[1-10]. 오버레이 멀티캐스트 기술에서 각각의 그룹 멤버들은 오버레이 멀티캐스트 트리에 의해서 각각 연결되어 있고, 멤버 관리 및 데이터 패킷 복사와 같은 멀티캐스트 기능들은 연결된 지점의 호스트들에 의해서 구현된다. 이 방식은 IP 멀티캐스트에서 발생하는 대부분의 문제를 해결해 주면서도 기존의 인터넷 구조에 쉽게 적용할 수 있기 때문에 멀티캐스트 기술의 도입을 쉽게 해줄 수 있다. 하지

만 오버레이 멀티캐스트 기술은 IP 멀티캐스트 기술에 비해서는 효율성이 떨어진다. 그 이유는 멀티캐스트 패킷 처리에 있어서 IP 멀티캐스트 기술에 비해 보다 큰 지연과 대역폭 손실이 발생 할 수 있으며 멀티캐스트 트리에 있어서도 안정성이 떨어질 수 있기 때문이다.

본 논문에서 우리들은 효율적이고 신뢰성 있는 멀티캐스트를 지원하기 위해서 두 가지의 기술을 채택하였다[11-12]. 하나는 ROMA에서 처음으로 소개된 loosely-coupled TCP connection을 수정하여 적용하였다[13]. 다른 하나는 IPv6 헤더의 Traffic Class에 새로운 서비스 타입을 정의하여 패킷의 종류에 따라 차별화된 서비스를 제공하는 것이다.

2. 관련 연구

오버레이 멀티캐스트 기술은 크게 고정 노드 방식과 동적

[†] 준 회 원 : 건국대학교 대학원 컴퓨터공학과(박사과정)

^{**} 정 회 원 : 건국대학교 컴퓨터정보통신학과 교수(교신저자)

논문접수 : 2006년 3월 28일, 심사완료 : 2006년 8월 28일

노드 방식으로 분류 될 수 있다. 고정 노드 방식은 노드들을 전체 인터넷에 전략적으로 분포 시켜놓은 후 어플리케이션의 요구 사항에 따라서 분포된 노드에 의해서 자율적으로 멀티캐스트 서비스를 수행할 수 있도록 멀티캐스트 트리를 구성하도록 하는 방식이다. 이 방식의 장점은 노드들이 고정되어 있기 때문에 구성된 멀티캐스트 트리는 안정적이고 어플리케이션에게 QoS를 쉽게 제공할 수 있다는 점이다. 반면에 멀티캐스트 서비스는 이러한 고정 노드를 통해서만 이루어져야 하기 때문에 효율성이 떨어지며, ISP에 의한 지원이 필요하고 네트워크의 병목 현상이 쉽게 발생한다는 단점이 있다. 동적 노드 방식은 고정 노드 방식과는 달리 오버레이 멀티캐스트 트리를 그룹 멤버간의 긴밀한 통신에 의해서 지능적으로 스스로 구성하도록 하는 방식이다. 이 방식에서 멀티캐스트 패킷 복사, 패킷 전달, 그룹 멤버 관리 및 이 외의 멀티캐스트 기능들은 모두 그룹 멤버들에 의해서 이루어진다. 멀티캐스트 그룹이 큰 경우에 그룹 멤버의 가입, 탈퇴는 빈번하게 발생할 것이며, 예상치 못한 네트워크 상황들이 발생할 수가 있다. 이러한 예상치 못한 변화들에 대해서 어떤 식으로 대처를 할 것인지가 동적 방식에서의 주요 이슈가 된다. 또 다른 중요한 이슈는 멀티캐스트 트리를 어떤 식으로 효율적으로 구성을 해서 효율적으로 그룹 통신을 할 것인지를 생각해야 한다는 점이다.

2.1 RON (Resilient Overlay Network)

RON은 네트워크 경로의 성능 저하와 불능으로부터 수초 내에 인터넷 어플리케이션들을 보호하고 복구하기 위한 구조이다. 광대역 라우팅 프로토콜(BGP)은 이와 같은 문제를 복구하는데 적어도 수분 정도 걸리는데 반해 RON은 이를 개선한 방법이다. RON을 사용하여 분산된 인터넷 어플리케이션에 참가하고 있는 노드들은 오버레이 네트워크를 형성하고 서로에게 협력적으로 패킷들을 전달한다. RON의 각 노드는 인터넷 상의 링크 품질, 패킷 손실 비율, 패킷 대기 시간, 사용 가능한 대역폭을 관측하고 이 정보를 다른 노드들에게 전달한다. 이것은 RON이 빠른 시간 내에 경로 실패에 대해 반응할 수 있게 하고, 성능 향상을 위한 특정 경로를 선택하게 한다[7-8].

2.2 Overcast

Overcast는 실시간 스트리밍 서비스를 위한 기법으로, 수천개의 클라이언트까지 지원할 수 있다. 세션 내의 클라이언트들의 생명 주기가 상대적으로 긴 응용 서비스에 적합하고, 네트워크 트래픽 상태 변화에 효율적으로 대처한다. Overcast는 소스로부터의 각 노드의 대역폭을 최대화하는 분산 트리를 구성하는 것과 효과적인 네트워크 토폴로지 이용에 초점을 맞추고 있다. 따라서 실시간 스트리밍 서비스를 위하여 대역폭의 효율적 사용을 목표로 하는 오버레이 멀티캐스트 기술이 된다[9].

2.3 ScatterCast

ScatterCast는 세션에 멀티캐스트 서비스를 제공하는

SCXs(ScatterCast Proxies)의 집합에 의해 서비스된다. 네트워크 상에 SCX라는 에이전트를 위치시키고 에이전트들간에 메쉬 구조의 네트워크를 형성한 후에 스페닝 트리를 구성한다. 이 때 에이전트는 각 해당 그룹의 클라이언트에게 서비스를 제공하며 또한 신뢰적 전송을 위한 에러 복구 등의 기능을 수행하게 된다. 이 기법은 소규모 그룹 통신에 있어서의 신뢰적인 브로드캐스팅 서비스를 목표로 한다[10].

3. 효율적인 오버레이 멀티캐스트 구조

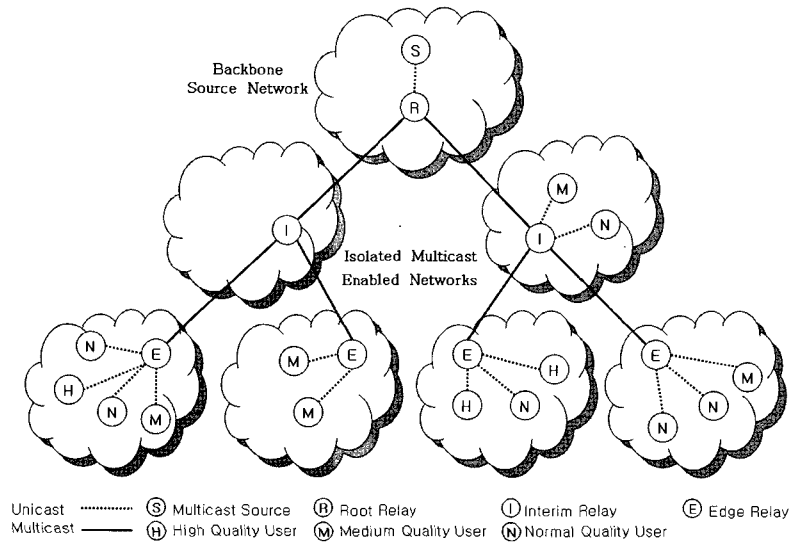
우리가 제안하는 플랫폼은 실시간 스트리밍 서비스를 위한 확장성과 신뢰성을 제공하는 오버레이 멀티캐스트 솔루션이다. 우리의 구조는 멀티캐스트를 지원하지 않는 네트워크를 통해 사용자들에게 스트리밍 데이터를 전달할 수 있으며 또한 데이터의 종류에 따라서 FEC(Forward Error Correction)와 차별화된 전송 서비스를 제공한다[14].

3.1 구성 요소

우리가 생각하는 기본 환경은 멀티캐스트 네트워크와 멀티캐스트를 지원하지 않는 네트워크가 혼합된 환경을 전체로 하고 있다. 현재 IP 멀티캐스트 네트워크의 전체 구성이 이루어지지 않은 상태에서 (그림 1)과 같이 오버레이 멀티캐스트 네트워크를 구축하였다. 멀티캐스트 데이터는 오버레이 멀티캐스트를 지원하는 릴레이를 통해서 전달되며 모든 릴레이는 사용자를 가질 수 있다. 멀티캐스트 데이터는 릴레이까지는 TCP를 사용한 유니캐스트로 전송되며 릴레이로부터 서브넷 안에 있는 사용자들에게는 UDP를 사용한 멀티캐스트로 전송된다. 릴레이는 멀티캐스트 데이터를 다른 릴레이에게 전달 하기 위한 버퍼와 서브넷의 사용자들에게 전달하기 위한 버퍼를 구별하여 사용한다. 또한 릴레이는 멀티캐스트 데이터를 사용자들에게 전달하기 전에 IPv6 헤더의 Traffic Class에 명시된 타입에 따라 FEC를 생성할 수도 있다.

Relay: 릴레이는 멀티캐스트 데이터를 전달하는 역할을 수행하는 장치로서 멀티캐스트 트리 구축의 구성요소가 된다. 우리가 제안하는 구조에서는 아래와 같이 3가지로 구분된다.

- Root Relay: 멀티캐스트 소스인 미디어 서버와 같은 네트워크에 있는 릴레이로서 멀티캐스트 트리의 루트가 된다. Root Relay는 멀티캐스트 데이터를 오버레이 멀티캐스트 헤더로 인캡슐레이션 시켜서 멀티캐스트 트리의 하위 노드들에게 전달을 한다.
- Interim Relay: 멀티캐스트 트리의 중간에 위치한 릴레이로 두 가지로 구분할 수 있다. 하나는 서브넷에 멀티캐스트 서비스를 희망하는 사용자가 없는 릴레이로 단순히 멀티캐스트 데이터를 다른 릴레이에게 전달을 해주는 기능을 수행하며 Traffic Class에 명시된 값에 따라 차별화된 전송이 가능



(그림 1) 제안된 오버레이 멀티캐스트 구조

하다. 다른 하나는 서브넷에 멀티캐스트 서비스를 제공받는 사용자가 존재하는 릴레이로 위에 설명한 기본 기능 이외에 별도의 기능을 수행한다. Traffic Class에 명시된 값에 따라 FEC를 포함한 데이터를 생성할 수 있으며, 서브넷의 멀티캐스트 가입자들에게 멀티캐스트 데이터를 전달한다.

- Edge Relay: Edge Relay는 멀티캐스트 트리의 끝에 위치한 릴레이로 서브넷에 존재하는 사용자들에게 멀티캐스트 데이터를 전달한다. Traffic Class에 명시된 값에 따라 FEC를 선택적으로 생성할 수 있다.

End-User: End-User는 멀티캐스트 데이터를 수신하는 단말로 수신한 멀티캐스트 데이터에서 오버레이 멀티캐스트 헤더 부분을 디캡슐레이션 시킨다. 또한 FEC의 포함 여부에 따라 오류 발생 시 자체적으로 에러 복구 작업을 수행한다. 우리가 제안하는 구조에서 End-User는 아래와 같이 3가지로 구분된다.

- High Quality User: 빠른 데이터 전송 서비스
FEC를 통해서 자체적인 에러 복구 작업 수행 가능
- Medium Quality User: 조금 빠른 속도의 데이터 전송 서비스
FEC를 통해서 자체적인 에러 복구 작업 수행 가능
- Normal Quality User: 일반적인 속도의 데이터 전송 서비스
FEC를 통해서 자체적인 에러 복구 작업 수행 불가능

3.2 특징

- 우리가 제안하는 구조는 다음과 같은 특징을 가진다.
- Reliability: IP 멀티캐스트가 제공하지 않는 신뢰적인 전송을 오버레이 멀티캐스트 네트워크를 통해서 제공한다

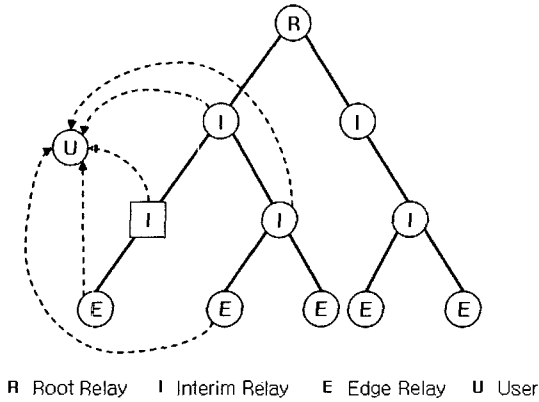
다. 신뢰적인 전송을 위해 loosely-coupled TCP 연결과 각각의 릴레이 사이에 유니캐스트 터널링을 사용하였고 서브넷 멀티캐스트 지역을 위해 선택적인 FEC 알고리즘을 채택하였다.

- Scalability: scalability를 지원하기 위하여 릴레이로 계층적인 멀티캐스트 트리를 구축하였으며, 서브넷에 릴레이가 없는 사용자를 위해서 진보된 멀티캐스트 그룹 가입 방법을 제시하였다.
- QoS: QoS를 지원하기 위하여 IPv6 헤더의 Traffic Class에 새로운 타입을 정의하였다. 새롭게 정의된 타입에 따라 서로 다른 버퍼를 사용하여 데이터 전송 속도를 차별화 하였다.
- Low Latency: 서로 다른 전송 속도의 버퍼를 사용함으로써 빠른 속도를 요구하는 멀티캐스트 데이터에 대하여 end-to-end 지연을 최소화 하였다. FEC를 사용함으로써 손실된 데이터의 재전송에 따른 지연을 감소시켰으며, 새로운 방법을 통해 멀티캐스트 트리 구성 시간을 감소시켰다.

4. 프로토콜

4.1 멀티캐스트 트리 구축

오버레이 멀티캐스트에서 멀티캐스트 트리는 릴레이를 중심으로 구성되며, 각각의 릴레이들은 NTP(Network Time Protocol)를 사용하여 시간 동기를 맞춘다. 멀티캐스트 서비스를 수신하길 희망하는 사용자는 해당 멀티캐스트 그룹에 대한 가입을 서브넷의 릴레이에게 요청한다. 만약 사용자가 속해 있는 서브넷에 릴레이가 존재하지 않는다면 사용자는 루트 릴레이에게 사용자로부터 가장 가까운 릴레이에 대한 정보를 요청한다. 이 요청 정보를 받은 루트 릴레이는 특정한 범위에 위치한 릴레이들에게 요청 정보를 전달한다. 릴



(그림 2) 멀티캐스트 그룹 가입 방법

레이들의 범위를 지정하는 방식은 사용자가 루트 릴레이에게 요청정보를 보내는 경로를 참조한다. 이를 위해 사용자는 루트 릴레이에게 요청정보를 보낼 때, 자신이 거쳐간 노드들에 대한 정보도 함께 전달한다. 루트 릴레이는 사용자로부터의 경로 정보에서 멀티캐스트 트리를 구축하고 있는 릴레이들이 속해 있는 네트워크 정보를 검색한다. 루트 릴레이는 검색된 정보 중에서 최하위 릴레이의 부모 노드 아래에 있는 모든 릴레이에게 사용자의 연결 요청 정보를 전달한다. 이 요청 정보를 받은 릴레이들은 동시에 사용자에게 응답 메시지를 전송하며, 사용자는 가장 짧은 응답 시간을 보이는 릴레이에 연결된다. (그림 2)는 서브넷에 릴레이가 없는 경우, 사용자가 멀티캐스트 그룹에 가입하는 방식을 보여준다. (그림 2)에서 사용자로부터 루트까지의 경로 정보에서 검색된 최하위 릴레이는 사각형으로 표현되었다. 이렇게 가입된 사용자는 멀티캐스트 데이터를 원격의 릴레이에서 유니캐스트로 전송 받는다.

사용자가 속한 서브넷에 릴레이가 없는 경우, 다른 네트워크의 릴레이가 대신 서비스를 해주는 것은 멀티플 유니캐스트 방식으로 동작하기 때문에 그 수가 증가 할수록 네트워크의 부하를 가중 시킬 수 있다. 네트워크의 부하를 방지하기 위해서 릴레이는 다른 네트워크의 서비스 제공에 대하여 제한을 할 필요가 있다. 릴레이는 네트워크 부하로 인하여 패킷 손실이 발생한 경우 다른 네트워크에 대한 서비스를 중지한다.

서브넷에 릴레이가 존재하지만 사용자가 희망하는 멀티캐스트 그룹에 릴레이가 가입되지 않은 경우도 위와 동일한 방법을 통해서 릴레이는 멀티캐스트 트리를 형성한다. 특정 후보 지역을 선정하여 릴레이로부터의 RTT를 측정하는 이 방법은 사용자에서부터 모든 릴레이까지의 RTT를 측정하는 방식보다 적은 지연 시간을 가진다. 또한 사용자에서 릴레이에 대한 정보를 관리하지 않기 때문에 사용자에게 부하를 주지 않는다.

4.2 QoS와 신뢰성 제공

Traffic Class 필드는 IPv6에 새롭게 추가된 필드로 IPv4

에서의 TOS(type of service)에 해당하는 영역이다. IPv6 헤더에서 Traffic Class는 4 bit의 version field 다음에 위치하고 있으며, 그 크기는 8 bit이다. 우리는 차별화된 멀티캐스트 패킷의 전송과 신뢰성 있는 데이터 전송을 효율적으로 수행하기 위하여 Traffic Class에 새로운 타입을 정의하였다. 차별화된 전송을 위해 두 개의 타입을 정의하였으며, 신뢰성 있는 전송을 위해 한 개의 타입을 정의 하였다. 현재 구현된 IPv6 모델에서는 Traffic Class의 값을 얻어오는 API를 지원하고 있기 때문에 어플리케이션에서는 손쉽게 QoS를 위한 세팅 값을 확인할 수 있으며, 향후 라우터에서 이 기능을 지원할 경우 보다 좋은 성능 향상을 기대할 수 있다.

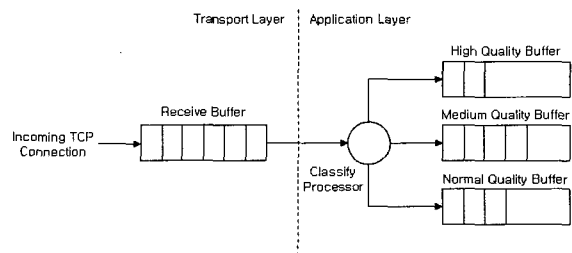
Traffic Class에 새롭게 정의한 타입은 다음과 같다.

- 10000000 : 차별화된 전송에 사용. 최상위 품질의 서비스를 요구하는 데이터로 다른 패킷들 보다 빠른 전송을 제공한다.
- 11000000 : 차별화된 전송에 사용. 중간 정도 품질의 서비스를 요구하는 데이터로 일반적인 패킷들 보다 조금 빠른 전송을 제공한다.
- 11100000 : 신뢰성 있는 전송에 사용. 서브넷 전송 시 FEC를 생성하여 수신자에서 자체 에러 복구가 가능하도록 한다.

차별화된 전송을 위한 타입들은 상호 배타적이지만 신뢰성 있는 전송을 위한 타입은 다른 타입들과 함께 설정될 수 있다.

(그림 3)은 차별화된 전송을 위한 구조를 나타내고 있다. 우리가 제안하는 구조는 ROMA에서 처음으로 소개된 loosely-coupled TCP connection을 수정한 모델이다. 릴레이 사이의 전송은 신뢰성 있는 TCP를 사용하며, 릴레이로부터 서브넷에 있는 사용자들까지의 전송에는 UDP를 사용한다. 하지만 실시간 스트리밍 서비스에서는 지연에 대한 문제가 중요하기 때문에 릴레이 사이의 통신에 TCP를 바로 사용하는 것이 아니라 릴레이 사이의 RTT 측정을 통해 지연 시간이 긴 경우는 UDP를 선택적으로 사용한다.

신뢰성 있는 TCP 연결을 통해 데이터를 전달받은 릴레이는 데이터의 처리를 위해 세 개의 버퍼를 사용한다. 릴레이는 Traffic Class에 정의된 타입에 따라 멀티캐스트 패킷을 세가지로 구분하여 High/Medium/Normal Quality Buffer

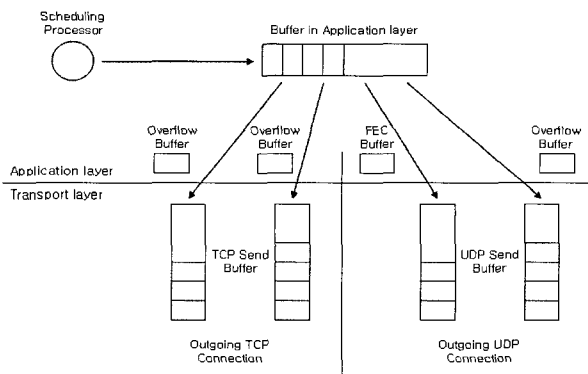


(그림 3) 차별화된 전송 방법

에 저장한다. Normal Buffer에는 일반적인 패킷들과 Traffic Class의 값이 정의된 타입과 일치하지 않는 패킷들이 저장된다. High Quality Buffer에 저장된 패킷들은 다른 버퍼에 저장된 패킷들보다 빠르게 처리되며, Medium Quality Buffer 저장된 패킷들은 Normal Buffer에 저장된 패킷보다 빠르게 처리된다.

릴레이에서 데이터 전송 방법은 두 가지로 구분된다. 하나는 멀티캐스트 데이터를 다른 릴레이에게 전송하기 위한 TCP 전송이며, 다른 하나는 서버넷에 있는 사용자들을 위한 멀티캐스트 전송이다. 하지만 긴 지연시간 때문에 릴레이들 간의 전송에 UDP를 사용한 경우, 릴레이들 사이의 전송에도 UDP 버퍼가 사용될 수 있다. 다른 릴레이를 위한 TCP 전송에는 해당 릴레이 별로 전송 버퍼와 Overflow Buffer가 존재한다. 서버넷을 위한 멀티캐스트 전송에는 신뢰성을 제공하기 위한 FEC Buffer와 단순 전달을 위한 Overflow Buffer가 존재한다. 릴레이는 멀티캐스트 데이터를 서버넷의 사용자들에게 전달하기 전에 Traffic Class에 명시된 타입에 따라 FEC 생성여부를 결정한다.

(그림 4)는 신뢰성 있는 데이터 전송을 위한 구조를 나타내고 있다. TCP는 신뢰적인 전송이기 때문에 손실된 패킷



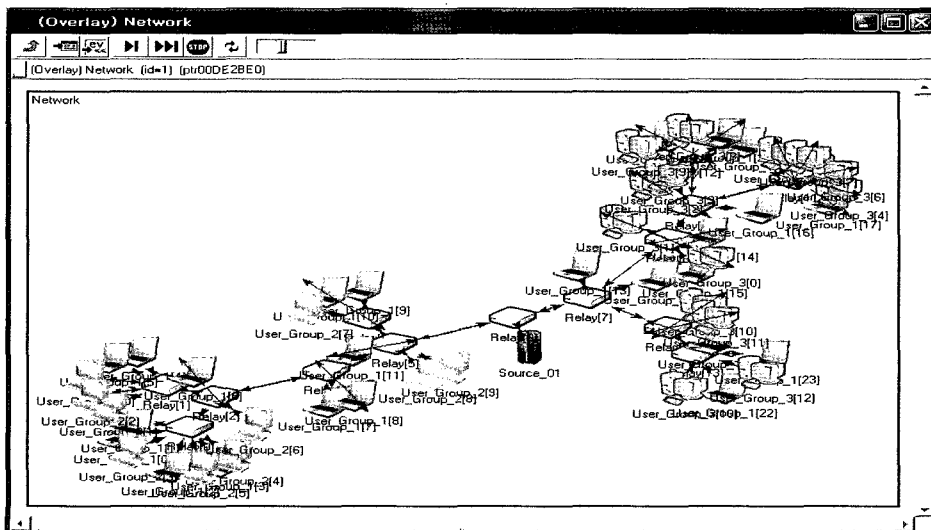
(그림 4) 차별화된 FEC 적용 방법

의 재전송이 가능하지만 실시간 스트리밍 서비스에서 재전송 방식의 사용은 지연이 크기 때문에 부적합하다. 멀티캐스트 서비스에는 이러한 지연을 줄이기 위해 수신자가 에러를 복구하는 FEC를 일반적으로 사용한다. 하지만 릴레이 간의 전송에 사용되는 TCP 자체가 신뢰적인 서비스를 제공하기 때문에 FEC의 사용은 큰 오버헤드를 발생시키며, 이는 성능 저하로 이어진다. 또한 서버넷에 전송되는 모든 UDP 패킷에 FEC를 적용하는 것도 오버헤드를 발생시킨다. 따라서 우리는 릴레이 간에는 TCP를 사용하고 서버넷에서는 Traffic Class에 명시된 값에 따라 선택적으로 FEC를 생성하여 멀티캐스트로 전송하는 방법을 채택하였다. 이 방식은 TCP와 UDP 그리고 FEC의 복합적인 사용으로 오버레이 멀티캐스트에서 실시간 스트리밍 서비스의 성능을 향상시켰다.

5. 성능 평가 및 결과

우리는 제안한 구조의 성능 평가를 위하여 작은 네트워크를 실제로 구성하였지만 광범위한 범위를 대상으로 성능 평가를 하기 위해서 네트워크 시뮬레이터들 중에 하나인 OMNeT++을 사용한 시뮬레이션을 수행하였다[15]. 우리는 오버레이 멀티캐스트 서비스를 위해 14개의 릴레이로 네트워크를 구성하였다. 릴레이와 릴레이들 사이 그리고 릴레이와 사용자들 간에는 양방향 전송이 가능하며 멀티캐스트 소스인 미디어 서버는 송신만 가능하게 설정하였다. (그림 5)는 OMNeT++을 사용하여 테스트베드 환경을 구축한 그림이다.

지연 측정을 위한 시뮬레이션에서 각 릴레이들 사이의 지연시간은 네트워크의 특성을 고려해서 5ms에서 20ms까지 차별해서 적용하였다. 멀티캐스트 패킷은 멀티캐스트 라우터 역할을 하는 릴레이를 통해서만 다른 네트워크로 이동이 가능하다. (그림 6)은 QoS에 대한 성능 평가를 위해 멀티캐스트 소스에서부터 각 릴레이까지의 지연 시간을 측정할 결



(그림 5) 성능 평가 모델

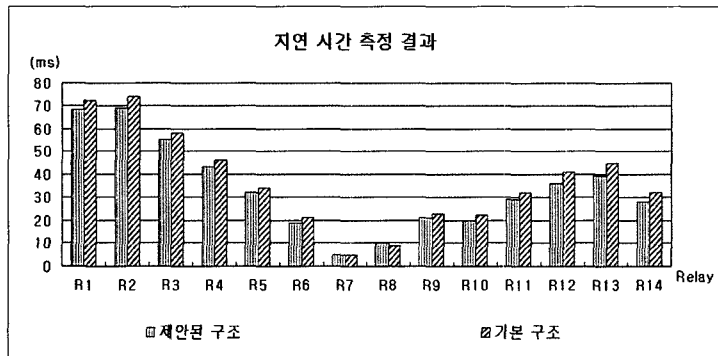
과이다.

(그림 6)의 측정 결과를 보면 멀티캐스트 소스와 같은 네트워크에 있는 루트 릴레이에 인접한 릴레이들 사이의 지연 시간은 우리가 제안한 QoS를 적용시킨 구조와 기본구조 사이에 차이가 나타나지 않는다. 하지만 루트 릴레이에서 거리가 멀어질수록 QoS를 적용시킨 우리의 구조가 기본구조보다 적은 지연 시간을 보이는 것을 확인 할 수 있다.

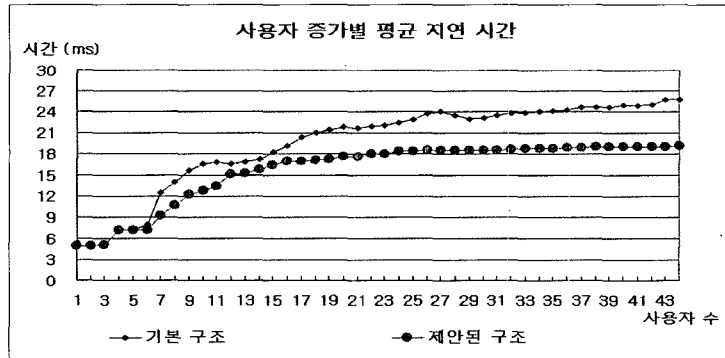
우리는 또한 확장성 평가를 위해 사용자 증가에 따른 전체 지연시간의 평균 지연시간을 측정하여 기본 오버레이 멀티캐스트 구조와 비교를 하였다. 실험의 신뢰성을 위해서 사용자 수를 점진적으로 증가 시키면서 반복적인 실험을 진행하였고 얻어진 결과의 평균값을 가지고 그래프를 작성하였다. (그림 7)은 사용자 증가에 따른 릴레이들의 평균 지연 시간을 측정한 결과이다. 사용자들의 서비스 요청 시간과

접속 위치는 랜덤하게 설정하여 측정하였다. 서비스가 전체 네트워크로 확장되어 가고 사용자 수가 증가 되어 갈수록 우리가 제안한 구조가 기본 구조보다 평균 지연 시간에서 우수한 성능을 나타내었다.

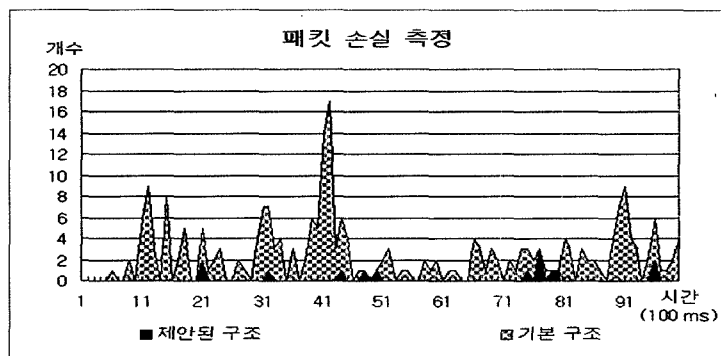
신뢰성 평가를 위해 우리는 전체 테스트베드에서 시간 증가에 따른 패킷 손실을 측정하였다. 시뮬레이션 결과에서 우리가 제안한 구조는 100% 패킷을 전송 받았지만 기본 구조에서는 상당한 양의 패킷 손실이 발생하였다. (그림 7)에서 기본 구조가 나타내는 그래프는 실제 손실된 패킷을 의미한다. 하지만 우리가 제안하고 있는 구조에서 나타나고 있는 패킷 손실에 대한 그래프는 실제 손실된 것을 의미하는 것이 아니라 실시간성을 고려했을 경우, 특정 시간 안에 도착되지 못한 패킷들을 의미한다.



(그림 6) 소스에서 각 릴레이까지의 지연 시간



(그림 7) 사용자 증가에 따른 평균 지연 시간



(그림 8) 시간 증가에 따른 패킷 손실

6. 결 론

본 논문에서 우리는 스트리밍 데이터를 위한 향상된 오버레이 멀티캐스트 구조를 소개하였다. 우리는 실시간 스트리밍을 위한 오버레이 멀티캐스트 구조의 문제점들을 검토하고 분석하였으며, 기존의 구조보다 효율적인 구조를 제안하였다. 우리의 구조가 기존의 구조와 다른 점들은 다음과 같다. 릴레이가 존재하지 않는 서버넷에 위치한 사용자가 멀티캐스트 트리에 가입하기 위한 효율적인 알고리즘을 제시함으로써 전체적인 지연의 감소와 scalability를 제공한다. 전송 부분을 릴레이 간의 전송과 서버넷에서의 전송의 두 부분으로 나누어서 접근을 하였다. 유니캐스트 터널링을 위해서 loosely coupled TCP connections을 사용하고 서버넷 멀티캐스트를 위해서는 선택적인 FEC를 사용하였다. IPv6의 Traffic Class에 새로운 타입을 정의하였으며, 그 타입에 따라 릴레이 간의 차별화된 전송과 서버넷 멀티캐스트 전송에 FEC 사용 여부를 결정하였다. 우리는 제안한 구조의 성능 평가를 위하여 네트워크 시뮬레이터들 중에 하나인 OMNeT++을 사용하여 몇 개의 모델을 대상으로 시뮬레이션을 수행하였다. 우리는 (그림 6)과 (그림 7) 그리고 (그림 8)의 시뮬레이션 결과를 통해서 우리가 제안하는 구조가 기존의 오버레이 멀티캐스트 구조보다 낮은 지연과 높은 신뢰성을 보장하는 것을 확인하였다. 따라서 우리의 접근은 QoS와 신뢰성을 제공하는 오버레이 멀티캐스트 구조를 위한 기본 메커니즘이 될 수 있다.

참 고 문 헌

[1] S. Banerjee, B. Bhattacharjee, and C. Kommareddy. "Scalable Application Layer Multicast," ACM SIGCOMM, 2002.

[2] J. Byers, J. Considine, M. Mitzenmacher, and S. Rost. "Informed Content Delivery Across Adaptive Overlay Networks," ACM SIGCOMM, Aug 2002.

[3] Y. Chu, S. G. Rao, and H. Zhang. "A Case For End System Multicast" ACM SIGMETRICS, 2000.

[4] P. Francis. "Yoid: Extending the Multicast Internet Architecture," 1999, White paper.

[5] J. Jannotti, D. K. Gifford, K. L. Johnson, M. F. Kaashoek, and J. W. O'Toole, Jr. "Overcast: Reliable Multicasting with an Overlay Network," USENIX Symp. on Operation Systems Design and Implementation, 2000.

[6] D. Pendarakis, S. Shi, D. Verma, and M. Waldvogel, "ALMI: An application level multi-cast infrastructure," the 3rd USENIX Symposium on Internet Technologies and Systems (USITS '01), 2001.

[7] David G. Andersen, Hari Balakrishnan, M. Frans Kaashoek, and Robert Morris, "The Case for Resilient Overlay Networks," HotOS 2001, pp 152-157, 2001.

[8] David G. Andersen, Hari Balakrishnan, M. Frans

Kaashoek, and Robert Morris, "Resilient Overlay Networks," 18th ACM SOSP, Oct., 2001.

[9] J. Jannotti, D. K. Gifford, K. L. Johnson, M. F. Kaashoek, and J. W. O'Toole, "Overcast: Reliable Multicasting with an Overlay Network" USENIX Symp. on Operation Systems Design and Implementation, 2000.

[10] Y. Chawathe, "Scattercast: An Architecture for Internet Broadcast Distribution as an Infrastructure Service," Ph.D. Thesis, University of California, Berkeley, Aug., 2000.

[11] R. Buskens, M. Siddiqui, S. Paul, "Reliable Multicast of Continuous Data Streams," Bell Labs Tech. Journal, 1997.

[12] S. Paul, K. Sabnani, J. Lin, S. Bhattacharria, "Reliable Multicast Transport Protocol (RMTP)," IEEE Journal on Selected Areas in Communications, Apr., 1997.

[13] G. Kwon and J. Byers. "ROMA: Reliable Overlay Multicast with Loosely Coupled TCP Connections," Technical Report BU-CS-TR-2003-015, Boston University, 2003.

[14] L. Rizzo and L. Vicisano, "RMDP: an FEC-based Reliable Multicast protocol for wireless environments", Mobile Computing and Communications Review, Volume 2, Number 2, 1998.

[15] OMNeT++ version 3.2, <http://www.omnetpp.org>

이 춘 성



e-mail : csrhee@cclab.konkuk.ac.kr

2001년 안양대학교 정보통신공학과

(공학학사)

2003년 건국대학교 대학원 컴퓨터 공학과

(공학석사)

2003년~현재 건국대학교 대학원 컴퓨터

공학과(박사과정)

관심분야: 차세대 네트워크, Overlay 멀티캐스트, Mobile IPv6, QoS 등

한 선 영



e-mail : syhan@cclab.konkuk.ac.kr

1977년 서울대학교 계산통계학과(학사)

1979년 한국과학기술원 전산학(석사)

1988년 한국과학기술원 전산학(박사)

1981년~현재 건국대학교 컴퓨터

정보통신학과 교수

1998년~1999년 미국 Maryland 컴퓨터과학과 객원교수

2004년 건국대학교 정보통신대학 학장

2005년 건국대학교 정보통신대학원 원장

관심분야: overlay 멀티캐스트, 멀티캐스트 보안, QoS, 차세대 네트워크 등



송정욱

e-mail : swoogi:hjhyun@cclab.konkuk.ac.kr

2000년 건국대학교 물리학과(이학사)

2004년 건국대학교 대학원 컴퓨터공학과
(공학석사)

2004년~현재 건국대학교 대학원 컴퓨터
공학과(박사과정)

관심분야: IPv6, Mobile IPv6, Network Mobility, Multihoming 등



최병욱

e-mail : buchoi@cclab.konkuk.ac.kr

2000년 배재대학교 컴퓨터공학과
(공학학사)

2002년 건국대학교 대학원
컴퓨터공학과(공학석사)

2002년~현재 건국대학교 대학원 컴퓨터
공학과(박사과정)

관심분야: SIP, 멀티캐스트 보안, overlay 멀티캐스트 등