라우터 기반의 신뢰적 멀티캐스트를 위한 버퍼 관리
(Buffer Management for the Router-based Reliable Multicast)

박 선 옥 † 안 상 현 ††
(Sunok Park) (Sanghyun Ahn)

요 약 다수의 수신자들이 하나의 송신자로부터 동일한 데이타를 받는 실시간 멀티미디어 서비스나 파일 전송 서비스 등이 일반화되면서 멀티캐스트와 같은 효율적인 그룹통신 메커니즘에 대한 관심이 증대되고 있다. 유니캐스트에 비해 적은 대역폭을 쓰며 그룹관리가 용이한 멀티캐스트의 효율성이 부각되면서, 확장성 및 신뢰성을 보장하기 위한 다양한 프로토콜들이 제안되었다. 최근에는 망의 트리 구조를 알고 있는 라우터를 기반으로 물리적인 트리를 구성하여 지역그룹내의 응답자(replier)를 선정하고 손실된 패킷에 대한 재전송을 처리해 주는 방법에 대한 연구가 진행되고 있다. 이 경우, 지역그룹내의 응답자는 망의 상태에 따라 임의적으로 선택되므로 멀티캐스트 그룹내의 모든 수신자들은 응답자가 될 가능성이 있다. 따라서 그룹내의 모든 수신자들은 손실 복구를 위한 버퍼를 유지하며 이 버퍼에 수신한 패킷들을 저장한다. 이는 그룹내 모든 수신자들이 불필요한 패킷을 항상 버퍼에 유지해야 하는 오버헤드를 가짐을 뜻한다. 따라서 본 논문에서는 더 이상 손실 복구 요청을 받지 않게 될 불필요한 패킷을 미리 판단하여 수신자들의 버퍼에서 패킷들을 지울 수 있게 함으로써, 불필요한 자원손실을 막는 방법을 제안한다. 제안된 방법에서는 LSM[1] 모델을 기반으로 응답자 선정 및 패킷 손실 복구를 하며, 지역그룹을 대표하는 삭제자(eraser)로부터의 ACK를 사용하여 불필요한 패킷을 판단하고 버퍼에서 해당 패킷을 지운다.

키워드 : 응답자, 삭제자, 버퍼 관리, 신뢰적인 멀티캐스트, 손실 복구

Abstract As services requesting data transfer from a sender to a multiple number of receivers become popular, efficient group communication mechanisms like multicast get much attention. Since multicast is more efficient than unicast in terms of bandwidth usage and group management for group communication, many multicast protocols providing scalability and reliability have been proposed. Recently, router-supported reliable multicast protocols have been proposed because routers have the knowledge of the physical multicast tree structure and, in this scheme, repliers which retransmit lost packets are selected by routers. Repliers are selected dynamically based on the network situation, therefore, any receiver within a multicast group can become a replier. Hence, all receivers within a group maintains a buffer for loss recovery within which received packets are stored. It is an overhead for all group receivers to store unnecessary packets. Therefore, in this paper, we propose a new scheme which reduces resource usage by discarding packets unnecessary for loss recovery from the receiver buffer. Our scheme performs the replier selection and the loss recovery of lost packets based on the LSM [1] model, and discards unnecessary packets determined by ACKs from erasers which represent local groups.

Key words : replier, eraser, buffer management, reliable multicast, loss recovery

1. 서 론

· 이 논문은 2000년도 서울시립대학교 학술연구조성비에 의하여 연구되었음.
† 정 회 원 : 한국전자통신연구원 통신프로토콜표준연구팀 연구원
sunok@etri.re.kr
†† 종신회원 : 서울시립대학교 컴퓨터과학부 교수
(corresponding author임)
ahn@venus.uos.ac.kr
논문접수 : 2002년 3월 15일
심사완료 : 2003년 3월 3일

하나의 송신자가 다수의 수신자들에게 동일한 데이타 전송을 하는 경우 멀티캐스트는 효율적인 수단을 제공해준다. 멀티캐스트를 사용하면 송신자가 동일한 데이타를 다수의 수신자들에게 반복해서 전송할 필요가 없기 때문에 송신자의 작업 부하가 감소하게 되며, 또한 동일한 데이타가 망 상의 경로를 중복해서 지나가는 경우를 최소화함으로써 자원을 보다 효율적으로 이용할 수 있

게 해준다. 이러한 멀티캐스트의 효율성이 부각되면서 다양한 멀티캐스트 관련 프로토콜들이 제안되었다[2]. 멀티캐스트와 관련된 이슈로는 확장성 문제와 신뢰적 전송 문제가 있을 수 있으며, 이들 문제에 대해서는 IETF의 RMT(Reliable Multicast Transport) 워킹그룹 등을 중심으로 활발히 연구가 이루어지고 있다[3~8].

IP 멀티캐스트는 최선의(best-effort) 방식을 사용하기 때문에 전송도중 패킷을 손실하거나 패킷을 수신하더라도 오류가 있을 수 있다. 따라서 멀티캐스트의 신뢰성을 보장하기 위해서는 별도의 손실 복구 방법이 요구되며, 대표적인 방법으로는 ARQ(Automatic Retransmission reQuest) 방식과 FEC(Forward Error Correction) 방식이 있다.

ARQ 방식은 오류탐지-재전송 요구-재전송에 걸친 3단계를 통해 손실 패킷을 복구한다. 수신자가 수신한 패킷의 순서번호(sequence number)를 기반으로 패킷 손실을 감지하거나 패킷 헤더의 검사합(checksum) 필드를 사용하여 비트 오류를 감지하게 되면, ACK(Acknowledge)나 NAK(Negative Acknowledge)를 통해 송신자에게 재전송을 요청한다. 재전송 요구를 받은 송신자는 해당 패킷을 다시 전송해줌으로써 손실을 복구한다.

FEC 방식[9]에서는 비트 오류 발생시, 송신자로부터의 재전송을 통해 손실을 복구하지 않고 수신자가 자체적으로 오류를 복구한다. 송신자는 보내고자 하는 데이타 k에 여분의 데이타를 함께 인코딩(encoding)하여 데이타 n을 전송한다. 그림 1과 같이 전송도중 비트 오류가 발생하더라도 n 데이타 중 k 데이타만 오류 없이 수신하면, 디코딩 과정을 통해 송신자가 보내고자 했던 데이타 k를 복구할 수 있다. 하지만 이 방식에서도 전송중에 잃어버린 패킷에 대한 손실 복구는 송신자의 재전송에 의해서 처리되어야만 한다.

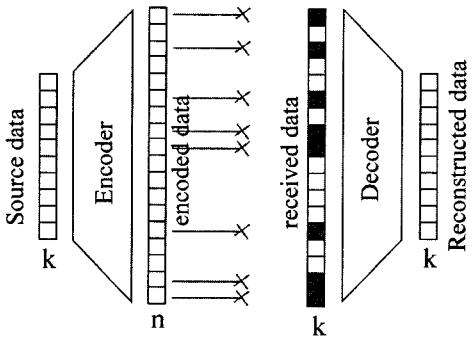

그림 1 FEC 방식

FEC 방식은 인코딩이나 디코딩을 처리하기 위한 오버헤드를 필요로 하며 또한 불필요한 여분의 데이타를 처음부터 전송해야 하므로, 불필요한 대역폭 낭비를 초래한다. 따라서 대부분의 신뢰적 멀티캐스트 방법들은 FEC 방식 대신 ARQ 방식을 사용하고 있다.

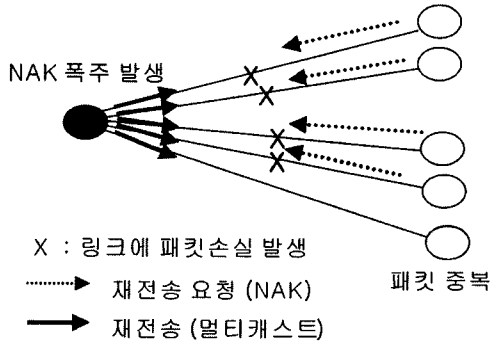

그림 2 멜티캐스트에서의 ARQ 방식

하나의 송신자가 다수의 수신자에게 멀티캐스트를 이용해서 데이타를 전송하는 경우, 기본적인 ARQ 방식은 그림 2에서와 같이 몇 가지 문제점이 있다. 패킷 손실을 탐지한 다수의 수신자들이 하나의 송신자에게 손실 복구를 위한 NAK 패킷을 모두 전송하면, 송신자측에서는 NAK 폭주(NAK implosion) 문제가 발생한다. 또한 손실 복구를 위한 재전송 패킷을 멀티캐스트를 이용하여 다시 전송하면, 손실 복구를 요청하지 않은 수신자도 재전송 패킷을 수신하게 되므로 노출(exposure) 문제가 생긴다. 이러한 문제들은 멀티캐스트 그룹 크기에 대한 제약을 하게 되며, 따라서 확장성 문제에 직면하게 된다.

이러한 확장성 문제를 해결하기 위해서, 전체 멀티캐스트 수신자 그룹을 지역그룹으로 나누어 지역그룹내의 응답자(replier)가 송신자를 대신하여 손실 복구를 수행하도록 하는 방법이 제안되었다[1]. 패킷 손실을 탐지한 수신자는 송신자에게 재전송 요청을 하지 않고 대신 지역그룹내의 응답자에게 요청을 하며, 응답자는 지역그룹으로만 손실 복구 패킷을 멀티캐스트함으로써 수신자 노출을 감소시킨다. 이 경우, 응답자는 지역그룹내의 모든 수신자들이 수신한 패킷에 대해서는 더 이상 손실 복구 요청을 받지 않게 될 것이므로 더 이상 버퍼에 해당 패킷을 저장하고 있을 필요가 없다. 특히 망의 상태에 따라 임의적으로 응답자가 선택되는 신뢰적 멀티캐스트 방법에서는 모든 수신자들이 응답자가 될 가능성이 있다. 따라서, 모든 수신자들은 손실 복구를 위한 버

퍼를 유지하고 있어야 하며, 수신한 패킷들을 모두 저장하고 있어야만 한다. 응답자의 버퍼에 저장되어 있는 불필요한 패킷들은 자원 낭비를 초래하며, 또한 응답자의 버퍼가 일정 크기로 제한되면 손실 복구를 요청한 수신자가 재전송 패킷을 수신할 때까지 소요되는 손실 복구 시간이 증가하게 된다.

대부분의 신뢰적 멀티캐스트 프로토콜에서 응답자는 수신자의 NAK 정보만을 기반으로 손실 복구를 하므로, 버퍼에 유지하지 않아도 되는 패킷을 판단하는데 있어서 어려운 점이 있다. 응답자가 불필요한 패킷을 판단하기 위해서는 지역그룹내의 모든 수신자로부터 ACK 패킷을 수신해야만 한다. 그러나 이 방법은 정확성을 제공하기는 하지만, ACK 폭주문제를 초래하며 불필요한 패킷을 판단하는데 오랜 시간이 걸리는 단점도 있다.

본 논문에서는 더 이상 손실 복구 요청을 받지 않게 될 패킷을 미리 판단하여 수신자들의 버퍼에서 패킷들을 지울 수 있게 함으로써, 불필요한 자원손실을 막는 방법을 제안한다. 이 기법은 LSM[1] 모델을 기반으로 응답자 선정 및 패킷의 손실 복구를 하며, 지역그룹을 대표하는 삭제자(*eraser*)로부터의 ACK를 사용하여 불필요한 패킷을 판단하고 버퍼에서 해당 패킷들을 지운다.

본 논문의 구성은 2절에서 관련 연구를, 3절에서 제안한 방법의 동작 방식을, 4절에서는 삭제자 선정 방법에 대해서 설명하고, 5절에서 실험을 통한 결과로부터 성능을 평가하고, 마지막으로 6절에서는 결론 및 향후 연구 과제를 제시한다.

2. 관련연구

2.1 신뢰적 멀티캐스트

신뢰적 멀티캐스트의 확장성 문제를 해결하고자 RMTP[10]에서는 전체 멀티캐스트 그룹을 지역그룹으로 나누어 지역그룹간의 계층 트리를 구성한다. 지역그룹내의 응답자가 송신자를 대신하여 손실 복구를 수행하며, 손실 복구를 처리해주지 못할 경우에는 상위 지역그룹의 응답자에게 손실 복구를 요청한다. 하지만 이 방법은 멀티캐스트 그룹 수신자들로 구성된 논리적인 계층 트리를 기반으로 지역그룹내의 응답자를 선정하고 손실 복구를 처리하므로, 자식-부모 관계가 바뀐 트리를 구성하게 되면 손실 복구 시간 측면에서 좋은 성능을 기대하기가 어렵다.

최근에는 망의 트리 구조를 알고 있는 라우터를 이용하여 손실 복구를 수행하는 연구가 진행되고 있다. 라우터를 이용한 손실 복구 방법을 제안한 LSM은 수신자가 손실된 패킷을 요청하면 라우터가 손실 복구 요청 패킷을 자신의 하위 링크나 상위 링크로 전달함으로써 손실 복구를 지원한다. 이러한 방법은 그룹 멤버들을 사용하여 지역그룹을 구성하는 방법보다 지역그룹 구성이 간단하고, 라우터가 손실 복구를 요청한 링크로 손실 복구 패킷을 전달하기 때문에 노출 문제를 감소시킨다.

2.2 신뢰적 멀티캐스트를 위한 버퍼 관리

복구요청을 처리하기 위해 저장되어 있는 패킷 중에서 불필요한 패킷을 판단하기란 그리 쉬운 일이 아니다. 버퍼 관리 기법을 적용하지 않았을 경우와 비교하여 신뢰적 멀티캐스트의 성능을 저하시키지 않아야 하며, 제어 메시지들의 폭주로 인한 확장성 문제도 고려해야 한다. 현재까지 제안된 대부분의 신뢰적인 멀티캐스트 프로토콜들은 응답자의 버퍼 관리에 대해서는 고려하지 않고 있다.

RMTP는 보조기억장치에 전체 멀티캐스트 세션 데이타를 저장하는 방법을 사용하고 있기 때문에 확장성을 제공하지 못하며, 버퍼에서 패킷을 지우기 위해 타이머를 사용하는 Search Party[11]는 타이머값 설정에 대한 언급을 하지 않고 있다.

신뢰적인 멀티캐스트의 버퍼 관리를 위해 새롭게 제시된 방법으로는 Gossip-Style Stability 프로토콜이 있다[12,13]. 이 방법에서는 그룹내의 모든 수신자들이 수신한 패킷을 안정된(stable) 패킷이라 하며 이러한 패킷을 감지하는 방법을 제안하고 있다. 동일한 시간을 갖는 단계들로 나누고, 각 단계동안, 그룹 수신자들은 자신의 상태 정보를 임의로 선택된 서브그룹에게 유니캐스트 방식으로 전송한다. 각 단계동안 다른 수신자로부터 상태 정보를 수신한 수신자는 자신의 현재 상태까지 반영한 새로운 상태 정보를 생성하며, 다음 단계동안 새로 생성한 상태 정보를 임의로 선택된 서브 그룹에게 전송한다. 몇 번의 단계를 반복하면, 임의의 수신자는 모든 그룹 수신자의 상태를 반영한 상태 정보를 수신하게 된다. 이 수신자가 불필요한 패킷을 결정해서 다른 수신자들에게 알려줌으로써 불필요한 패킷을 버퍼에서 지우게 한다. 하지만, 이 방법은 송신자가 데이타 전송 전에 전체 멀티캐스트 그룹 수신자를 미리 알고 있어야만 정확한 버퍼관리를 할 수 있다는 단점이 있으며, 또한 사전에 그룹 멤버를 알고 있는 경우라 하더라도 전체 그룹크기에 대한 확장성 문제와 수신자들이 자신의 상태를 반영하기 위해 상태 정보를 매번 수정해야 하는 오버헤드를 지닌다. 이에 반해 본 논문에서 제시한 방법은 사전에 전체 그룹 수신자들을 알고 있지 않아도 된다.

3. 삭제자를 이용한 버퍼 관리

3.1 개요

신뢰적인 멀티캐스트를 제공하기 위해 논리적인 계층 트리를 사용하는 프로토콜들은 손실된 패킷을 복구하기 위해 그룹 멤버들간에 계층적인 트리를 구성한다. 그러나 논리적인 트리는 물리적인 트리와 다를 수 있기 때문에 자식-부모 관계가 바뀔 수 있으며, 이 경우 지역 오류 복구가 올바르게 동작하지 않는 문제가 있다. 따라서 본 논문의 응답자 구성 방법과 동작 원리는 LSM과 같은 물리적인 망을 이용한 손실 복구 방법을 기반으로 한다.

LSM은 라우터를 이용한 물리적인 계층 트리를 구성하여 라우터의 하위에 위치한 수신자중에서 응답자를 선정하고, 응답자가 송신자를 대신하여 손실 복구를 수행하도록 하는 방법이다. 그룹내의 수신자들은 응답자로서 동작하기 위해 라우터에 요청을 하며, 라우터는 최소 비용을 가진 수신자를 응답자로 선정한다. 라우터는 응답자의 IP 주소 대신에 응답자로 향하는 링크 정보를 유지하고 손실 복구 요청 패킷을 응답자 링크로 전달한다. 상위 링크로 향하던 손실 복구 요청 패킷이 하위 링크로 향하게 되는 지점에 있는 라우터를 TP(Turning Point)라고 하며, 손실 복구 요청 패킷이 TP를 통과할 때 라우터는 자신의 IP 주소와 패킷이 수신된 링크 정보를 패킷에 기록한 다음에 응답자에게 전달한다. 손실 복구 요청 패킷을 수신한 응답자는 재전송을 수행할 수 있으면 손실 복구 요청 패킷에 기록된 링크 정보를 기록하여 재전송 패킷을 생성한 뒤, 패킷에 기록되어 있던 TP의 IP 주소로 캡슐화 하여 TP로 터널링한다. 라우터는 캡슐화된 재전송 패킷을 수신하면 패킷을 역캡슐화 하여 링크 정보를 추출한 후, 재전송된 멀티캐스트 패킷을 해당 링크로 멀티캐스트 함으로써 지역 복구를 처리한다.

그림 3과 그림 4는 LSM에서의 손실 패킷 요청 및 응답자의 재전송 과정을 보여준다.

그러나 LSM은 지역그룹내의 모든 수신자들이 수신한 패킷을 여전히 응답자의 버퍼에 저장하고 있으므로 자원손실을 초래한다. 따라서 본 논문에서는, 지역그룹내의 손실 복구를 처리하는 응답자의 버퍼를 관리하기 위해 수신자들중 최대 비용을 갖는 수신자를 삭제자로 선정한다. 응답자는 지역그룹에서 선정된 삭제자로부터의 ACK를 기반으로 불필요한 패킷을 판단하며, 해당 패킷들을 자신의 버퍼에서 제거함과 동시에 지역그룹내 다른 수신자들에게도 알려준다. TP마다 하나의 응답자와 하나의 삭제자가 선정된다.

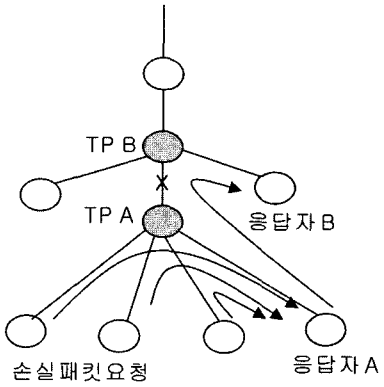

그림 3 손실 패킷 요청

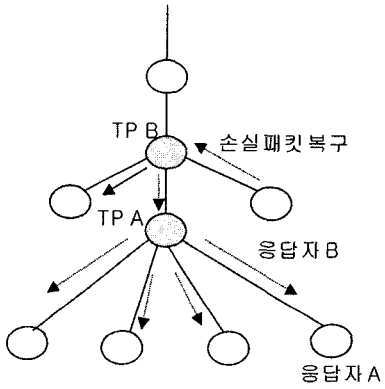

그림 4 손실 패킷 복구

3.2 망 구성 요소

· 라우터

지역그룹내 모든 수신자들로부터 받은 CANDIDATE [Cost, Null] 메시지를 기반으로 최소 비용을 가진 수신자를 응답자로 선정하며, 하위 모든 링크로부터 받은 메시지를 통합한 CANDIDATE[Min_Cost, Max_Cost] 메시지를 상위 라우터에게 전송한다. 이 메시지를 수신한 상위 라우터에서는 하위 링크로부터 받은 정보들을 기반으로 최대 비용을 가진 수신자를 하위 TP를 위한 삭제자로 선정한다.

· 응답자

지역그룹내 수신자들의 패킷 손실에 대한 복구를 처리하며, 응답자 선정 방법이나 패킷에 대한 손실 복구 방법은 앞서 설명한 LSM[2]을 기반으로 한다. 삭제자로부터의 ACK를 기반으로 불필요한 패킷을 판단하고 지역그룹내의 모든 수신자들이 버퍼에서 해당 패킷을 지우도록 지역그룹내에서 멀티캐스트로 알려준다.

· 삭제자

응답자 구성시에 지역그룹내의 최대 비용을 가진 수신자를 삭제자로 선정한다. 응답자는 지역그룹내 모든 수신자들로부터 ACK를 수신해서 불필요한 패킷을 판단해야 하지만, 이러한 방법은 불필요한 패킷을 판단하는데 걸리는 시간이 길며 ACK 폭주 문제를 초래하여 확장성 측면에서 좋지 못한 결과를 가져온다. 따라서 본 논문에서는, 가장 성능이 좋지 않은 수신자가 수신한 패킷은 다른 수신자들도 수신하였을 것이라 가정하고 삭제자를 기반으로 응답자의 버퍼에서 불필요한 패킷을 제거하게 한다.

그룹 멤버들은 주기적으로 CANDIDATE 메시지를 상위 링크로 전송해서 TP를 재구성하고 응답자와 삭제자가 재선정되도록 한다.

3.3 손실 복구 방법

응답자는 지역그룹을 대표하는 삭제자로부터의 ACK를 기반으로 불필요한 패킷을 판단하므로, 손실 복구 요청을 직접 처리해주지 못할 수도 있다. 즉, 불필요한 패킷으로 판단되어 버퍼에서 이미 지워버린 패킷에 대한 복구 요청은 해당 응답자가 처리해줄 수 없다. 이 경우 상위 응답자에게도 해당 패킷이 없을 확률이 높으므로 송신자에게 직접 손실 복구를 요청한다.

4. 삭제자 선정

삭제자의 ACK 정보만을 기반으로 불필요한 패킷을 판단하므로 삭제자 선정 방법은 아주 중요하다. 삭제자 선정시 사용되는 수신자 비용은 대역폭(bandwidth), 홉수(hop count), 링크 지연 시간(link delay) 등 여러 가지 요소를 반영할 수 있다. 망의 상태에 따라 트리는 재구성되며 응답자와 삭제자도 재선정된다.

4.1 메시지 종류

삭제자 선정을 위해 정의된 제어 메시지들은 다음과 같다:

CANDIDATE[Cost, Null] 메시지
: 수신자가 자신의 비용값을 라우터에게 전송하는데 사용하는 메시지

CANDIDATE[Min_Cost, Max_Cost] 메시지
: 하위 링크로부터 수신한 CANDIDATE 메시지를 기반으로 최소 비용과 최대 비용값을 상위 라우터로 알려주기 위해 라우터가 생성하는 메시지

ERASER_SELECT 메시지
: 응답자에게 주기적으로 ACK 메시지를 보내는 삭제자를 선정하는데 사용되는 메시지

ERASER_SELECT_REPLIER 메시지
: ERASER_SELECT 메시지와 마찬가지로 삭제자 선정을 위해 사용되나, 응답자로 동작하는 수신자를 상위 TP의 삭제자로서 동작하게 하고자 할 때 사용하는 메시지

ERASER_GIVEUP 메시지
: 망의 상태에 따라 새로운 삭제자가 선정되면, 기존의 삭제자에게 더 이상 삭제자로서의 역할을 수행하지 않도록 알릴 때 사용하는 메시지

ERASER_ACK 메시지
: 삭제자가 응답자에게 자신이 수신한 최대 패킷 순서번호를 알려 주는데 사용하는 ACK 메시지

STABLE 메시지
: 응답자가 삭제자의 ACK 메시지를 기반으로 불필요한 패킷을 판단해서 지역그룹내 다른 수신자들에게 알려주는데 사용하는 메시지

4.2 응답자/삭제자 선정 알고리즘

라우터가 하위 모든 링크로부터 CANDIDATE[Cost, Null] 메시지를 수신하면, 새로운 CANDIDATE[Min_Cost, Max_Cost] 메시지를 생성해서 상위 링크로 해당 메시지를 전송한다. CANDIDATE 메시지를 수신한 각 라우터들은 하위 링크의 상태 정보를 저장하기 위한 테이블을 구성하고 유지한다. 이 테이블을 기반으로 응답자와 삭제자가 선정되며, 선정된 응답자와 삭제자에 대한 정보도 함께 테이블에 저장된다. 그림 5는 TP인 라우터 A에서의 삭제자 선정 과정을 보여주고 있다.

각 링크에 있는 숫자는 링크 비용을 의미하며, 링크 ID는 왼쪽에서부터 일련번호로 붙여진다고 가정한다. 4명의 수신자들은 각각 CANDIDATE[Cost, Null] 메시지를 라우터 A에게 보내며, 라우터 A는 이들 정보를 통합해서 CANDIDATE[Min_Cost, Max_Cost] 메시지를 상위 라우터 B에게 전송하고 가장 작은 비용을 가진 하위 링크를 응답자 링크로 선정한다. 라우터 B는 하위 두 링크로부터 CANDIDATE 메시지를 수신하면 Min_Cost 값을 비교해서 가장 작은 Min_Cost 값을 가진 링크를 응답자 링크로 선정하고 응답자 링크를 제외한 다른 하위

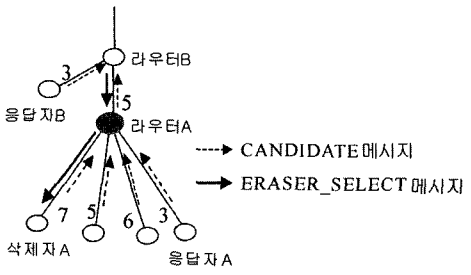

그림 5 삭제자 선정 과정

표 1 라우터 A의 상태 정보 테이블

	Link_ID	Min_Cost	Max_Cost
Replier_	4	3	Null
Eraser_			
Max_	1	7	Null

링크로는 ERASER_SELECT 메시지를 전송한다.

라우터 A는 상위 링크로부터 ERASER_SELECT 메시지를 수신하면, 저장되어 있는 상태 정보 테이블을 바탕으로 삭제자를 선정한다.

표 1은 각 라우터가 가지고 있는 상태 정보 테이블을 보여주고 있으며, Replier_ 레코드에는 선정된 응답자에 대한 링크 정보 및 비용이 기록된다. Max_ 레코드에는 하위 링크 중 최대 비용을 가진 링크에 대한 정보가 기록되며, 삭제자가 선정되면 Eraser_ 레코드에 삭제자에 대한 정보가 기록된다.

라우터가 상위 링크로부터 ERASER_SELECT 메시지를 수신하면, 이미 삭제자가 설정되어 있는지 Eraser_ 레코드를 먼저 검사한다. 삭제자가 설정되어 있지 않으면, Max_ 레코드에 기록되어 있는 링크로 ERASER_SELECT 메시지를 전달하고 Eraser_ 레코드에 Max_ 레코드를 복사한다. 만약 삭제자가 이미 설정되어 있으면, Replier_ 레코드에 기록되어 있는 응답자 링크로 ERASER_SELECT_REPLIER 메시지를 전송한다.

표 1은 그림 5의 예에서 TP인 라우터 A의 상태 정보 테이블을 보여준다. 라우터 A가 상위 링크로부터 ERASER_SELECT 메시지를 수신하면, Max_ 레코드에 기록되어 있는 1번 링크로 메시지를 전달하고 Eraser_ 레코드의 링크 ID는 1, Min_Cost와 Max_Cost는 7과 Null값으로 각각 기록한다.

각 라우터는 두 개 이상의 응답자와 삭제자 링크를 가질 수 없으며, 손실 복구 요청 패킷을 수신하면 응답자 링크로 해당 패킷을 전송해서 응답자가 손실 복구를 처리할 수 있도록 한다.

하위 TP는 ERASER_SELECT 메시지를 수신하면 아래와 같은 방법으로 삭제자를 선정한다. 그림 6은 TP에서의 가능한 하위 링크 타입을 보여주고 있으며, 각 링크 타입에 따라 서로 다른 동작에 의해 삭제자가 선

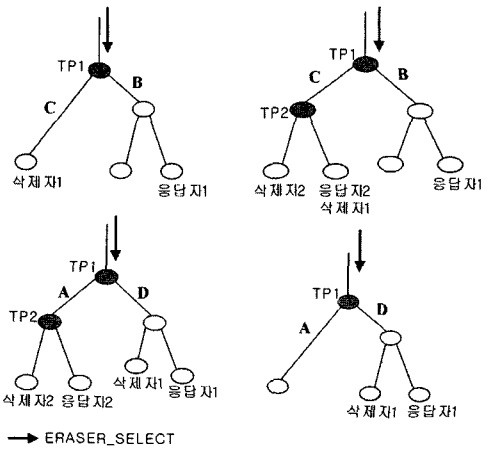

그림 6 TP에서의 가능한 하위 링크타입

정된다. 삭제자1과 응답자1은 TP1의 삭제자와 응답자를 각각 의미하며, 상위 TP로부터 ERASER_SELECT 메시지를 수신함으로써 해당 TP의 삭제자가 선정된다.

각 라우터에서 수신하게 되는 CANDIDATE 메시지와 저장되어 있는 테이블의 상태 정보를 기반으로 링크 타입이 결정되며, 링크 타입에 따라 서로 다른 제어 메시지들이 하위 라우터로 전송된다. 표 2는 서로 다른 링크 타입에 따라 라우터가 전송하게 되는 메시지를 보여준다.

4.3 메시지 수신시 라우터의 동작 방식

삭제자 선정을 위해 여러 종류의 메시지들이 사용되며, 하위 라우터에서 이들 메시지를 수신했을 때 취하게 되는 동작은 다음과 같다:

CANDIDATE 메시지 수신

라우터의 상태 정보 테이블의 Replier_ 레코드의 Min_Cost 보다 수신한 메시지의 Min_Cost가 더 크고, 메시지의 Max_Cost가 Null이 아니면 메시지를 수신한 링크로 ERASER_SELECT 메시지를 전송

ERASER_SELECT 메시지 수신

상태 정보 테이블의 Eraser_ 레코드가 설정되어 있으면, Replier_ 레코드에 설정되어 있는 응답자 링크로 ERASER_SELECT_REPLIER 메시지를 전송하고, 그렇지 않으면 Max_ 레코드에 설정되어

표 2 링크타입에 따라 전송되는 메시지

비용 (Cost)		Max : ○ Min : X		Max : X Min : X		Max: X Min : ○	Max : ○ Min : ○
Max_Cost == Null?		Yes	No	Yes	No	무관	무관
결과	링크 타입	C′	C	A′	A	B	D
	전송메시지	없음	ERASER_SELECT	없음	ERASER_SELECT	없음	없음

있는 최대 비용을 가진 링크로 ERASER_SELECT 메시지를 전달

ERASER_SELECT_REPLIER 메시지 수신

설정된 응답자 링크로 ERASER_SELECT_REPLIER 메시지를 전달

ERASER_GIVEUP 메시지 수신

설정된 응답자 링크로 메시지를 전달

ERASER_GIVEUP2 메시지 수신

설정된 삭제자 링크로 메시지를 전달

그림 7에서는 이러한 제어 메시지를 수신한 라우터의 동작을 알고리즘을 통해 상세히 기술하고 있다. 지역그룹내의 응답자와 삭제자는 CANDIDATE 메시지를 기반으로 라우터가 선정한다. 따라서 라우터가 아닌 응답자를 기준으로 비용을 계산하면 선정된 삭제자의 비용이 최대가 아닐 수도 있다. 특히 응답자와 삭제자가 같은 링크에 존재하면 더 그럴 것이다. 따라서 그림 6에서 타입 D인 경우는 발생하지 않도록 알고리즘을 작성했다.

```
// CANDIDATE 메시지 수신
    If (replier_min_cost > new min_cost)
       { If (replier_max_cost != Null)
            send ERASER_SELECT to replier link;
         update RST[Change new link to replier link];
         if (new_max_cost != Null)
            send ERASER_GIVEUP2 to new link;
       }
    Else
       { if (new_max_cost != Null)
            send ERASER_SELECT to replier link;
         update RST;
       }
    Send aggregated CANDIDATE [replier_min_cost, max_min_
                    cost] to upstream
// ERASER_SELECT 메시지 수신
    If (max_max_cost != Null)
       { if (eraser_iface > 0)
            send ERASER_SELECT_REPLIER to max link;
         update RST;
       }
    Else
       { if (eraser_iface != max_iface)
            send ERASER_GIVEUP to eraser link;
         update RST [change max link to eraser link];
         forward ERASER_SELECT to eraser link;
       }
// ERASER_SELECT_REPLIER 메시지 수신
       Forward ERASER_SELECT_REPLIER to replier link
// ERASER_GIVEUP 메시지 수신
       Forward ERASER_GIVEUP to replier link
// ERASER_GIVEUP2 메시지 수신
       Send ERASER_GIVEUP to eraser link
       Eraser_ record initiate (in RST)
```

그림 7 제어 메시지를 수신한 라우터의 동작 알고리즘

4.4 삭제자 선정 예

TP에서 삭제자를 선정하는 과정을 구체적인 예를 통해 알아본다. 그림 8은 4개의 라우터와 5개의 수신자로 구성된 망의 예를 보여주고 있으며, 링크에 쓰여진 숫자는 패킷 전송을 위한 비용을 나타낸다. 링크 ID는 왼쪽에서부터 일련번호를 부여하며, CANDIDATE 메시지는 왼쪽 링크에서부터 차례로 수신된다고 가정한다.

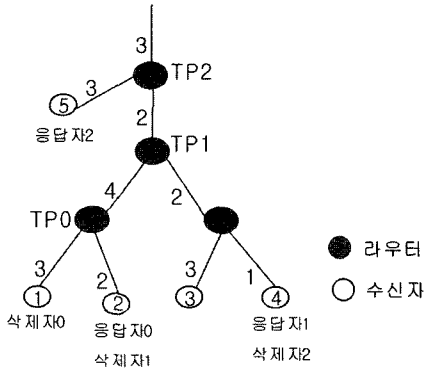

그림 8 삭제자 선정 예

각 수신자들은 CANDIDATE[Cost, Null] 메시지를 라우터로 보내며, 모든 하위 링크로부터 이 메시지를 수신한 라우터는 필요한 정보를 자신의 상태 정보 테이블에 저장한 후 최소 비용을 가진 링크를 응답자 링크로 설정한다. 이 과정을 통해 수신자 2와 수신자 4가 응답자로 선정된다. 응답자를 선정한 라우터는 모든 하위 링크에 대한 최소 비용과 최대 비용을 CANDIDATE[Min_Cost, Max_Cost] 메시지를 이용하여 상위 라우터로 다시 전송한다.

TP1에서, 먼저 1번 하위 링크로부터 CANDIDATE[6, 7] 메시지를 수신하게 되면 1번 링크를 응답자 링크로 선정함과 동시에, 상태 정보 테이블의 Replier_와 Max_ 레코드의 Min_Cost와 Max_Cost에 각각 6과 7의 값을 기록한다. 이후 2번 하위 링크로부터 CANDIDATE[3, 5] 메시지를 수신하면, 상태 정보 테이블에 기록되어 있는 비용과 새 CANDIDATE 메시지의 비용을 비교한다.

1. CANDIDATE[3,5] 메시지를 링크2로부터 수신하면,
   - ▶ Replier_ Min_Cost(6) > new Min_Cost(3) 인지 비교 (만족)
     - ① 링크2를 응답자 링크로 재선정
     - ② 링크1의 Max_Cost가 NULL이 아니므로, 링크1로 ERASER_SELECT 메시지 전송

2. TP0에서 ERASER_SELECT 메시지를 수신하면,
   ▶ 삭제자 링크가 설정되어 있는지 검사 (설정되어 있지 않음)
   ① 링크1이 최대링크로 설정되어 있으므로, 링크1로 ERASER_SELECT 메시지를 전달
   ② ERASER_SELECT 메시지를 수신한 수신자 1이 TP0의 삭제자로서 동작
3. TP1에서는 CANDIDATE[5, 8] 메시지를 상위 TP2로 전송

TP2에서도 마찬가지로, 두 하위 링크로부터 수신한 CANDIDATE[5, 8] 메시지와 CANDIDATE[3, Null] 메시지를 이용하여 하위 TP1의 삭제자를 선정한다. 이러한 과정은 멀티캐스트 트리의 루트에 도달할 때까지 모든 라우터에서 반복적으로 행해지며, TP마다 하나의 응답자와 하나의 삭제자가 선정된다.

5. 실험결과

기존의 신뢰적 멀티캐스트 프로토콜을 기반으로 제안한 버퍼 관리 메커니즘을 동작시켰을 때 어떠한 성능을 나타내는지 알아보기 위해 NS v.2[14] 시뮬레이터를 사용했으며, 실험을 위해 사용한 망 구성은 그림 9와 같다. 각 링크의 대역폭은 1.5Mbps이고 링크지연 값은 10ms로 가정했으며, 송신자는 1ms마다 1024byte 패킷을 CBR(Constant Bit Rate)로 전송한다. 그룹에 참여하는 수신자 수는 16이며 실험을 위해 각 링크의 패킷 손실률을 10%로 하였고 수신자의 비용은 패킷 손실률을 기반으로 했다.

본 실험에서는 본 논문에서 제시한 삭제자 기반 버퍼 관리 기법에 대해 버퍼 관리 기법을 적용하지 않았을 경우와 비교하여 손실 패킷 평균 복구시간에 얼마나 많은 오버헤드를 갖는지 살펴본다. 또한, 만약 그룹내 수신자들이 일정한 버퍼 크기를 가진 시스템이라면, 본 논문에서 제안한 방법이 버퍼 관리 기법을 적용하지 않았을 경우와 비교하여 어떤 성능을 보이는지 살펴본다. 따라서, 버퍼관리를 고려하지 않은 무한버퍼를 가진 LSM과 일정한 크기로 버퍼를 고정시켰을 때의 LSM, 그리고 본 논문에서 제안한 방법을 적용한 LSM의 성능을 비교한다.

제한된 버퍼를 가진 LSM에서는 손실 복구를 위한 패킷을 더 이상 버퍼에 저장하지 못하게 되면, 가장 오래된 패킷부터 하나씩 버퍼에서 삭제한다. 그림 10에서는 버퍼크기에 따른 손실 복구 시간을 보여준다.

본 논문에서 제시한 버퍼관리 기법을 적용한 경우와 무한버퍼를 가진 기존의 LSM을 비교해보면, 무한 버퍼를 가진 기존의 LSM은 손실 패킷 복구시간이 0.907초로 일정하나, 버퍼관리 기법을 적용한 경우에는 0.907초보다 크게 나타난다. 이는 삭제자로부터의 ACK를 기반으로 수신자의 버퍼에서 불필요한 패킷을 판단하고 버퍼에서 해당 패킷을 삭제하므로 삭제된 패킷에 대해 손실 복구 요청을 받게 되는 경우, 상위 응답자로부터 손실 패킷에 대한 복구가 이루어지기 때문에 발생하는 현상이다. 하지만, 수신자의 버퍼가 12K이상인 경우에는 무한버퍼인 경우와 비교할 때 사용 버퍼는 작지만 손실 복구 시간은 많은 차이가 없음을 볼 수 있다. 그룹내 수신자들이 일정한 크기의 버퍼를 가지는 경우, 본 논문에서 제안한 방법이 버퍼 관리 기법을 적용하지 않았을 경우와 비교하여 어떤 성능을 가지는지 알아보기 위해, 제한된 버퍼를 가진 LSM과 삭제자를 가진 LSM의 성능을 비교해보았다. 버퍼크기가 7K인 경우를 예를 들면, 제한된 버퍼를 가진 LSM의 경우 2.486초의 평균 손실 복구 시간을 가지나, 삭제자를 가진 LSM의 경우에는 1.224초의 평균 손실 복구 시간을 가진다. 따라서, 동일한 버퍼크기를 가진 시스템에서는 본 논문에서 제안한

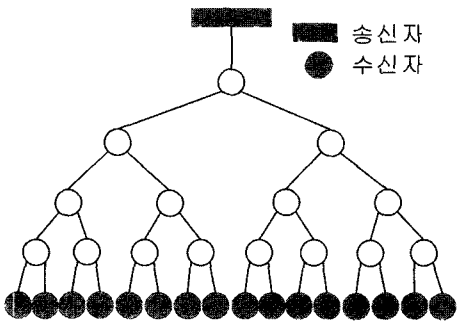

그림 9 실험을 위한 망 구조

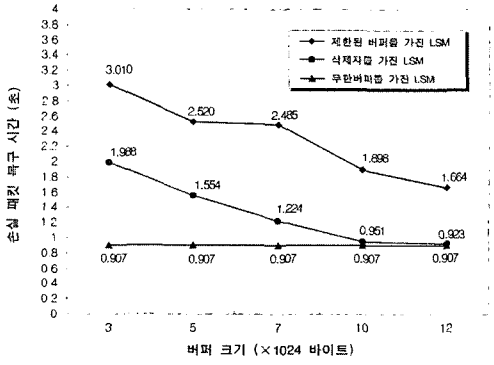

그림 10 버퍼크기에 따른 손실 패킷 복구 시간

버퍼관리 기법을 적용한 경우 평균 손실 복구 시간이 줄어듦을 알 수 있다. 또한 수신자들이 12K 이상의 버퍼를 가진 경우에는 무한 버퍼를 가진 LSM만큼의 성능을 가짐을 알 수 있다. 제안한 버퍼관리 기법은 기존의 LSM과 비교하여 동일한 성능을 제공하기 위해 더 적은 버퍼를 사용한다는 장점이 있으나, 삭제자를 선정하기 위한 추가적인 제어 메시지들이 필요한 단점이 있다.

6. 결론 및 향후 연구과제

본 논문에서는 라우터를 기반으로 지역그룹내의 모든 수신자가 수신했을 패킷을 감지하는 방법을 제안하였다. 이 기법은 LSM 모델을 기반으로 하며, 지역그룹을 대표하는 삭제자로부터의 ACK를 사용하여 응답자가 불필요한 패킷을 판단하고 버퍼에서 해당 패킷들을 지울 뿐만 아니라, 그룹내의 다른 수신자들에게 알림으로써 불필요한 자원 손실을 막는다. 또한 응답자의 버퍼가 일정 크기로 제한된다면 손실 복구를 요청한 수신자가 재전송 패킷을 수신할 때까지 소요되는 손실 복구 시간이 단축됨을 실험을 통해 보여주었다. 제안된 방법에서는 송신자가 전체 그룹 멤버에 대한 정보를 알고 있을 필요가 없으며, 그룹 멤버가 빈번히 바뀌는 환경에서도 삭제자를 이용해서 버퍼 관리를 할 수 있다. 멀티캐스트 확장성이나 신뢰성뿐만 아니라 멀티캐스트 보안과 관련된 점도 고려되어야 하며, LSM 외에 다른 신뢰적 멀티캐스트 방법을 기반으로 삭제자를 이용한 버퍼 관리를 적용시키는 방안도 향후 연구되어야 할 과제로 남아있다.

참 고 문 헌

[1] Papadopoulos, C. and Parulkar, G., "An Error Control Scheme for Large-Scale Multicast Applications," *Proceedings of INFOCOM 1998.*

[2] Deering, S., "Host Extensions for IP Multicasting," RFC1112, January 1989.

[3] Floyd, S., et al., "A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing," Proceedings of ACM Sigcomm '95, September 1995.

[4] Levine, B. and Garcia-Luna-Aceves, J., "A comparison of reliable multicast protocols," ACM Multimedia Sys., August 1998.

[5] Pingali, S., et al., "A Comparison of Sender-initiated and receiver-initiated Reliable Multicast Protocols," SIGMETRICS 1994.

[6] Speakman, T., et al., "Pragmatic General Multicast Transport Protocol Specification," IETF draft-speakman-pgm-spec-02.txt, August 1998.

[7] Cain, B., et al., "Generic Router Assist Building Block Motivation and Architecture," IETF draft-ietf-rmt-gra-arch-01.txt, March 2000.

[8] Yavatkar, R., et al., "A Reliable Dissemination Protocol for Interactive Collaborative Applications," Proceedings of ACM Multimedia '96, 1996.

[9] Gemmell, T., et al., "The use of Forward Error Correction in Reliable Multicast," IETF draft-ietf-rmt-info-fec-01.txt, October 2001.

[10] Paul, S., et al., "RMTP: A Reliable Multicast Transport Protocol for High-Speed Network," Proceedings of the Tenth Annual IEEE Workshop on Computer Communications, September 1995.

[11] Costello, A. and McCanne, S., "Search party: Using randomcast for reliable multicast with local recovery," Proceedings of IEEE INFOCOM '99, March 1999.

[12] Guo, K. and Rhee I., "Message Stability Detection for Reliable Multicast," *INFOCOM 2000.*

[13] Reness, T. and Minsky, Y., et al., "A gossip-style failure detection service," Proceedings of Middleware '98, 1998.

[14] McCanne, S. and S. Floyd, NS (Network simulator), http://www-nrg.ee.lbl.gov/ns, 1995.

박 선 옥

1999년 서울시립대학교 전산통계학과 (전산학) 학사. 2002년 서울시립대학교 컴퓨터·통계학과 컴퓨터네트워크 석사. 2002년 2월~현재 한국전자통신연구원 통신프로토콜표준연구팀 연구원 관심분야는 Reliable Multicast, Internet Protocol, SIP, VoIP

안 상 현

정보과학회논문지 : 정보통신
제 30 권 제 1 호 참조