

신뢰적 멀티캐스트를 위한 타이머 기반 버퍼 관리

(Timer-based Buffer Management for Reliable Multicast)

안상현[†] 김영민^{**} 권영호^{***}
(Sanghyun Ahn) (Youngmin Kim) (Youngho Kwon)

요약 멀티캐스트 트래픽의 효율적 전달을 위해서는 멀티캐스트 라우팅과 신뢰성 있는 전송 방식이 요구된다. 신뢰성 있는 전송은 손실 패킷에 대한 재전송을 필요로 하며, 이를 위해 전송된 패킷은 재전송 버퍼에 저장되어 있어야만 한다. 따라서 재전송 버퍼의 관리 문제가 중요하며, 본 논문에서는 재전송 버퍼에 얼마나 많은 패킷을 저장할 것인가 하는 문제를 다룬다. 본 논문에서 제안하는 방식인 TBM (Timer based Buffer Management)은 멀티캐스트 트리 상의 응답자와 수신자들 사이의 NAK를 이용해서 얻어진 타이머 값을 기준으로 적정량의 버퍼를 유지하며, 동적인 네트워크 상황에 적용한다. 실험을 통해 TBM이 오류 상황이나 네트워크의 크기 등에 상관없이 버퍼를 능동적으로 관리하는 것을 보인다.

키워드 : 신뢰적 멀티캐스트, 타이머 기반 버퍼 관리, 지역 복구, LMS

Abstract In order to deliver multicast traffic efficiently, multicast routing and reliable transmission mechanisms are required. The reliable delivery implies that lost packets must be retransmitted, which in turn requires that transmitted packets be stored in a retransmission buffer. Therefore how to manage a retransmission buffer is important and, in this paper, we try to solve the problem of how many packets should be maintained in the buffer. Our proposed scheme, the timer based buffer management (TBM), maintains only necessary amount of buffer based on the timer value calculated from the NAKs between the replier and receivers on a multicast tree and can adjust to the dynamic network conditions. By performing simulations, we show that TBM manages the buffer efficiently regardless of the error situation, network size, and so on.

Key words : Reliable multicast, Timer based buffer management, Local recovery, LMS

1. 서론

여러 수신자에게 데이터를 보내야 할 경우 멀티캐스트[1,2] 전송방식을 사용하면 망 전체의 트래픽 양을 줄이면서 동일한 데이터를 요청한 여러 사람에게 전송할 수 있다. 하지만 데이터 전송중 오류가 발생했을 경우 이를 복구하는 문제에 있어서는 여러 가지 고려되어야

할 이슈가 있다. 일대일 통신과는 달리 수신자가 여럿이기 때문에 하나의 송신자와 다수의 수신자들이 데이터를 정확하게 주고 받았는지 확인하는 과정에서 많은 문제들이 발생할 수 있다.

멀티캐스트에서는 수신자가 여럿이기 때문에 같은 패킷에 대한 재전송 요청이 한꺼번에 송신자에게 보내질 수 있으며, 이로 인해 송신자측에서 NAK 폭주(NAK implosion) 문제가 발생할 수 있다. 또한 송신자가 NAK에 대한 응답으로 손실 복구를 위한 재전송 패킷을 멀티캐스트 방식으로 전송하면, 손실 복구를 요청하지 않은 수신자도 재전송 패킷을 수신하게 되는 노출(exposure) 문제를 야기한다. 이러한 문제점들은 멀티캐스트 그룹 크기가 클수록 더 심각한 영향을 나타낸다.

이러한 문제점을 해결하기 위해 망의 구조를 알고 있는 라우터를 이용한 손실 복구방법이 연구되고 있다. 라

· 본 연구는 한국과학재단 목적기초연구(R04-2001-00054) 지원으로 수행되었음

† **통신회원** : 서울시립대학교 컴퓨터과학부 교수
(corresponding author)
ahn@venus.uos.ac.kr

** **학생회원** : 서울시립대학교 컴퓨터.통계학과
blhole@venus.uos.ac.kr

*** **비회원** : 서울시립대학교 컴퓨터.통계학과
yhwon95@venus.uos.ac.kr

논문접수 : 2002년 6월 24일

심사완료 : 2003년 3월 27일

우터의 물리적인 망 정보를 이용하여 손실 복구 방법을 제안한 LMS[3]는 수신자가 NAK를 보내 손실된 패킷을 요청하면 라우터에서 NAK를 자신의 하위 링크나 상위 링크로 전달하여 손실 복구를 지원한다. LMS는 라우터를 기반으로 하며 지역 그룹내의 응답자(replier)를 사용하여 손실 복구를 수행하므로 확장성을 제공하고 효율적인 손실 복구를 수행한다.

LMS의 경우 패킷 손실을 탐지한 수신자는 송신자에게 재전송 요청을 하는 것이 아니라 지역 그룹내의 응답자에게 요청하며, 응답자는 지역그룹으로만 손실 복구 패킷을 멀티캐스트 한다. 이 경우 지역 그룹내의 모든 수신자들이 수신한 패킷에 대해서는 더 이상 손실 복구 요청을 받지 않게 될 것이므로 응답자는 더 이상 버퍼에 해당 패킷들을 저장하고 있을 필요가 없다. 특히 네트워크의 상태에 따라 유동적으로 응답자가 선택되는 신뢰적 멀티캐스트 방법에서는, 손실 복구를 위해 사용되는 버퍼를 모든 수신자들이 가지고 있어야 하며 수신한 패킷들을 저장하고 있어야 한다. 이때 응답자의 버퍼에 저장되어 있지만 지역그룹의 모든 수신자들이 이미 수신하여 더 이상 유지할 필요가 없는 패킷들을 지니고 있는 경우 손실이 초래되며, 또한 메모리의 크기는 한계가 있으므로 일정 크기로 응답자의 버퍼를 제한한다면 복구할 패킷이 응답자의 버퍼에 존재하지 않으므로 인해 손실 복구를 요청한 수신자가 재전송 패킷을 수신할 때까지 소요되는 손실 복구 시간(loss recovery time)이 증가할 수 있다.

응답자의 버퍼를 관리하기 위해 불필요한 패킷을 정확하게 판단하려면 지역 그룹내의 모든 수신자들이 응답자에게 ACK 패킷을 전송해야 한다. 그러나 이 방법은 전송되는 대부분의 패킷에 대해 ACK 패킷이 전송되므로 ACK 폭주 문제가 생길 수 있다. 따라서 대부분의 신뢰적 멀티캐스트 프로토콜에서는 응답자가 수신자의 NAK 정보만을 기반으로 손실 복구를 처리하기 때문에 버퍼에 유지할 필요가 없는 안정된(stable) 패킷[4]을 판단하기 어렵다.

본 연구에서는 수신자의 NAK에 패킷 손실 발견시간을 삽입하여 응답자에게 전송하면 응답자는 그 값과 NAK의 전송시간을 적용하여 계산된 타이머 값으로 응답자와 수신자들의 버퍼에서 더 이상 손실 복구 요청을 받지 않을 것으로 예상되는 패킷을 판단하고 지울 수 있게 한다. 이 방법을 이용하여 신뢰적 멀티캐스트를 서비스하는 네트워크 장비에서 실행되는 다른 응용(Application)들에게 할당되지 못하고 패킷 복구를 위해 고정적으로 할당되어 낭비되는 메모리 자원을 줄이는 방법

을 제안한다. 이 기법에서의 응답자 선정 및 패킷의 손실 복구는 LMS 모델을 기반으로 한다. 멀티캐스트를 구현하기 위해서는 네트워크 계층에서의 멀티캐스트 라우팅과 트랜스포트 계층에서의 신뢰적 멀티캐스트 분야로 연구가 진행되고 있으며, 본 연구에서는 신뢰적 멀티캐스트 중에서도 NAK를 기반으로 지역 회복(local recovery) 기법을 사용할 때 언제든지 응답자의 역할을 담당할 수 있는 수신자들의 버퍼를 어떻게 하면 효율적으로 관리할 수 있는가에 대한 해결책을 제시한다.

2. 관련 연구

SRM[5]은 송신자에게서만 재전송을 받을 수 있기 때문에 버퍼를 유지하지 않으며 RMTP[6]에서는 버퍼 관리 개념 없이 보조기억장치에 전체 멀티캐스트 세션 데이터를 저장하는 방법을 사용하고 있기 때문에 확장성을 제공하지 못한다. Search Party[7]에서는 메모리에서 패킷들을 삭제하기 위해 간단한 타이머(course-grain timeout)를 사용할 수 있다고 제안하고 있지만 이에 대한 자세한 내용을 다루지 않고 있다.

Gossip-Style Stability 프로토콜[4][8]은 안정된 패킷을 알아내기 위해 모든 멀티캐스트 수신자들이 일정한 시간 간격으로 나누어진 단계마다 그룹에서 임의로 선택된 가십 부집합(gossip subset)으로 자신의 현재 정보를 보낸다. 각 단계동안 자신의 패킷 수신 정보와 다른 수신자로부터 받은 패킷 정보를 조합하여 다음 단계에 새로 만들어진 가십 부집합으로 전송한다. 이러한 단계를 반복하면 임의의 수신자는 모든 그룹 수신자의 상태가 반영된 상태 정보를 수신하게 되는데 이 수신자가 안정된 패킷을 판단하여 멀티캐스트 그룹상의 모든 또는 일부 수신자들에게 알려 줌으로써 불필요한 패킷을 버퍼에서 지우게 한다. 그러나 이 방법은 송신자가 데이터 전송 전에 멀티캐스트 그룹 멤버를 미리 알고 있어야만 사용할 수 있으며, 알지 못하는 경우에는 확률적으로 결정된 단계 수만큼만 반복한 뒤 불필요한 패킷을 판단하므로 안정된 패킷에 대한 신뢰적인 감지가 어려운 문제가 있다. 또한 전체 그룹 크기가 커질수록 안정된 패킷을 판단하는데 오랜 시간이 걸리는 확장성 문제와 수신자들이 자신의 상태를 반영하기 위해 상태 정보를 매번 수정해야 하는 오버헤드를 지닌다.

복구 요청을 처리하기 위해 응답자의 버퍼에 저장되어 있는 패킷 중에서 불필요한 패킷을 판단하는 문제는 수신자들의 메모리 자원을 절약할 수 있는 중요한 방법이지만, 이를 효과적으로 관리하는 것은 쉬운 일이 아니며 현재까지 제안된 대부분의 신뢰적 멀티캐스트 프로

토콜들은 응답자의 버퍼 관리에 대해서는 고려하지 않고 있다.

3. 타이머를 이용한 버퍼 관리

3.1 손실된 패킷의 복구

버퍼 관리 기법을 살펴보기 전에 멀티캐스트 데이터 전송중 패킷 손실이 발생했을 때 이를 감지한 수신자가 재전송을 요청하여 복구받는 과정을 기술한다. 그림 1은 물리적인 멀티캐스트 트리의 일부로서 LMS 방식을 이용한 예이다. 응답자는 지역 그룹을 대표하여 복구요청에 대한 재전송을 담당하고 TP(turning point)는 일반적인 라우터의 기능에 추가적으로 수신자에게서 받은 재전송 요청 메시지(NAK)를 응답자에게 유니캐스트로 보내주는 기능과 응답자의 재전송 메시지를 지역 그룹으로 멀티캐스트(directed multicast)하는 기능을 가진다.

멀티캐스트 트리 상에서 TP 아래로는 하나의 응답자가 있으며, 멀티캐스트 그룹의 수신자들이 가입과 탈퇴를 이행하거나 응답자에서 재전송 요청이 높은 빈도로 발생할 경우 기존에 응답자가 아니었던 수신자가 응답자의 역할을 할 수 있다. 따라서 응답자와 수신자 모두 패킷 손실 복구를 위한 자신의 버퍼를 유지하고 있어야 한다.

송신자가 1에서 5까지의 일련번호(sequence number)를 갖는 5개의 패킷을 전송하는 상황에서 한 수신자가 3, 4번 패킷을 수신하지 못했을 경우, 이를 감지한 수신자는 3, 4번 패킷에 대한 재전송 요청과 2번 패킷을 받은 시간(t_2)을 NAK에 포함시켜 송신자 쪽으로 보낸다. TP가 이 NAK를 수신하면 해당 응답자에게 전달해 주며, 응답자는 3, 4번 패킷을 TP에게 보내고 TP는 지역 그룹으로 멀티캐스트 하여 복구를 수행한다.

3.2 타이머를 이용한 버퍼 관리

송신자가 전송한 패킷 수만큼 버퍼의 크기가 무한정

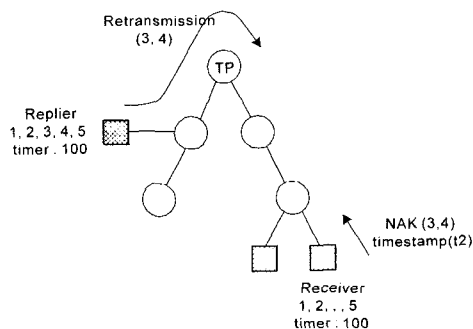


그림 1 응답자와 송신자간 송수신 메시지

으로 늘어난다면 시스템의 자원은 고갈될 것이다. 그러므로 더 이상 재전송 요청이 없는 패킷을 판단하여 버퍼에서 제거해주는 작업이 필요하다. 본 연구에서는 응답자의 타이머를 이용하여 이 작업을 수행한다. 앞으로 기술할 TBM(timer-based buffer management) 기법을 이용하면 멀티캐스트 그룹 멤버 수에 상관없이 응답자의 재전송 실패수, 응답자 및 수신자의 평균 버퍼 사용량을 최적화된 값에 근사하게 할 수 있다.

타이머를 관리하기 위한 전제조건으로 모든 수신자들은 각 패킷이 도착할 때마다 그 패킷을 받은 시간을 기록하며, NTP(Network Time Protocol) 등을 이용하여 멀티캐스트 그룹 멤버간의 시간을 동기화 함으로써 응답자가 버퍼에서 적절한 시간에 패킷을 삭제할 수 있게 한다.

그림 2에서 수신자의 버퍼에는 패킷의 일련번호와 그 패킷을 수신한 시간이 저장되어 있으며, 송신자가 1, 2, 3, 4, 5의 일련번호를 가지는 패킷을 전송했을 때 지역 응답자는 모두 올바르게 수신했지만 지역 그룹의 한 수신자가 1, 2번 패킷을 받은 후 5번 패킷을 받아 3, 4번 패킷을 수신하지 못했을 경우를 보여준다. 수신자가 5번 패킷을 수신하면 3, 4번 패킷이 손실된 것을 감지하고 2번 패킷을 수신한 시간(LastReceivedTime)을 포함시킨 NAK 패킷을 LMS의 방식으로 응답자에게 보낸다.

실제 일련번호와 다른 순서로 패킷이 수신된 경우 패킷 손실로 판단하여 NAK를 보내는 상황이 일어날 수 있으며, 이런 경우에 대비해 일정 시간을 기다린 후에도 앞쪽 패킷이 도착하지 않을 경우 손실로 판단한다.

NAK 패킷의 메시지 형식은 그림 3과 같으며 지역 그룹으로만 멀티캐스트 방식으로 전송된다. 'NAK low'는 손실된 패킷의 시작번호(그림 2의 예에서는 3)이고 'NAK high'는 손실된 패킷의 끝번호(그림 2의 예에서는 4)이며, 'LastReceivedTime'은 NAK low 바로 전에 수신한 패킷의 시간(그림 2의 예에서는 2번 패킷을 수신한 시간)이다.

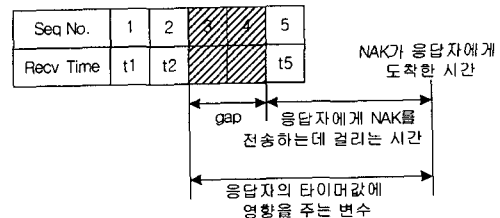


그림 2 타이머 계산 방식을 보여주기 위한 수신자의 버퍼

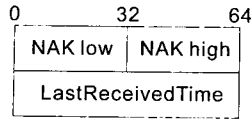


그림 3 NAK의 메시지 형식

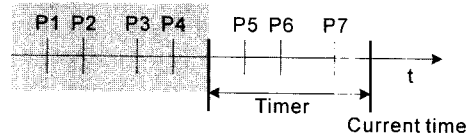


그림 4 버퍼에서 오래된 패킷의 삭제

NAK를 수신한 응답자는 NAK에 포함된 정보인 LastReceivedTime과 NAK가 응답자에 도달한 시간과의 차이(NakReceivedTime)를 타이머에 반영하며, 이 값은 수신자가 패킷 손실을 감지하고 재전송을 요청하여 응답자에게 도착하는데 걸리는 평균적인 시간을 얻어내기 위한 정보로 사용된다. 응답자의 입장에서는 이 시간동안만 버퍼에 패킷을 저장하고 있고 그 후에는 재전송 요청이 없을 것으로 판단하여 해당 패킷을 삭제한다. 이 방법은 멀티캐스트 그룹 멤버의 수가 늘어나더라도 재전송을 위해서는 반드시 필요한 NAK만을 이용하여 버퍼를 관리하므로 관련 연구들에서 지적되어 왔던 확장성 문제를 해결할 수 있다.

TCP의 RTT 계산에 사용되는 방법을 응용한 식 (1)은 응답자측에서 타이머를 계산하는 방법으로 α 값을 이용하여 최근에 계산된 타이머 값을 어느 정도 비율로 반영할지 정할 수 있다. 패킷이 송수신되기 전단계에서의 타이머 초기값은 송신자와 수신자간 RTT값의 두 배를 사용하며 응답자 이외의 모든 수신자들은 이 값을 타이머 값으로 이용한다.

$$timer_n = \alpha * timer_{n-1} + (1 - \alpha) * NakReceivedTime, \quad 0 < \alpha < 1 \quad (1)$$

만일 응답자가 NAK에 해당하는 패킷을 가지고 있지 않아 재전송을 해주지 못할 때에는 타이머 값을 늘려주고 LMS의 방법을 이용하여 상위의 응답자나 송신자에게서 복구받는다. 식 (2)는 타이머 값을 늘려주는 방법으로 복구 실패시 값을 이용하여 한 배에서 두 배까지 조정하여 사용할 수 있다.

$$timer_n = \beta * timer_{n-1}, \quad 1 < \beta < 2 \quad (2)$$

모든 수신자가 응답자의 역할을 할 수 있는 환경에서 응답자가 그 역할을 중단하고 수신자로 돌아가는 경우가 있는데 이 경우에는 더 이상 NAK를 수신하지 않게 되며 복구를 위한 큰 버퍼를 필요로 하지 않는다. 또한 응답자라 하더라도 일정시간동안 NAK를 수신하지 않으면 패킷 전송이 원활이 되고 있는 상황이기 때문에 버퍼의 크기를 조금씩 줄여나간다. 식 (3)은 일정시간동안 NAK를 수신하지 않았을 때 타이머 값을 줄이는 방법을 보여준다. 단, 타이머 값은 식 (3)에 의해서 초기값 ($2 * RTT$)보다 작아지지는 않는다.

$$timer_n = \gamma * timer_{n-1}, \quad 0 < \gamma < 1, \quad timer_n > 2 * RTT \quad (3)$$

응답자가 NAK를 받으면 식 (1)에 의해서 타이머 값을 적절히 조정하고 만일 재전송이 실패하면 식 (2)에 의해 타이머 값을 증가시키며, 일정시간동안 재전송 요청을 받지 않으면 타이머 값을 감소시킨다. 각 식에서 사용되는 α, β, γ 값을 어떻게 설정하면 더 좋은 성능을 낼 수 있는지에 대해서는 실험 결과를 통해 자세히 살펴보도록 하겠다.

위의 식 (1), (2), (3)에 의해 타이머 값이 결정되면 일정한 주기마다 버퍼를 검사하여 현재 시간에서 타이머를 뺀 값보다 작은 수신 시간을 갖는 패킷을 지우며 이러한 버퍼 삭제 방법은 응답자와 수신자 모두에 대해 동일하게 적용된다. 그림 4에서 회색으로 표시된 P1, P2, P3, P4는 삭제될 패킷이다.

4. 실험 결과

타이머를 이용한 버퍼 관리 기법의 성능을 분석하기 위해서 NS v.2[9]를 사용했으며, 기본적인 신뢰적 멀티캐스트 기능은 LMS를 기반으로 했다. TBM의 확장성을 보이기 위해 그림 5의 세 네트워크에 대해 실험했으며, 모든 링크의 대역폭은 1.5Mbps, 링크지연 시간은 5ms으로 동일하고, 송신자는 1024바이트의 패킷을 40ms마다 CBR(Constant Bit Rate)로 총 500초 동안 전송하며, 수신자의 한 버퍼 크기는 1024바이트이다.

패킷의 손실을 발생시키는 방법은 중요한 성능 분석 요소로 사용되며 전체 링크 중에서 동일한 확률로 4개의 링크를 선택해서 일정한 시간(5ms) 동안 패킷 손실을 발생시킨다. 라우터의 버퍼가 가득 차서 패킷 전달이 원활하지 못한 것이 패킷 손실의 원인이므로 한 번에 여러 개의 연속된 패킷이 손실되며 실험에서는 40개의 패킷중 2개, 5개, 10개의 패킷을 연속적으로 손실시켰다.

표 1은 그림 5의 (a)네트워크에서 40개 중 5개의 패킷을 손실시키는 오류 모델에 대해, 식 (1)의 α 값을 0.3, 0.5, 0.7로, 식 (2)의 β 값을 1.3, 1.5, 1.7로, 식 (3)의 γ 값을 0.5, 0.7, 0.9로 각각 대입시켜서 얻어낸 응답자 및 수신자의 평균 버퍼 사용량, 응답자의 복구 실패

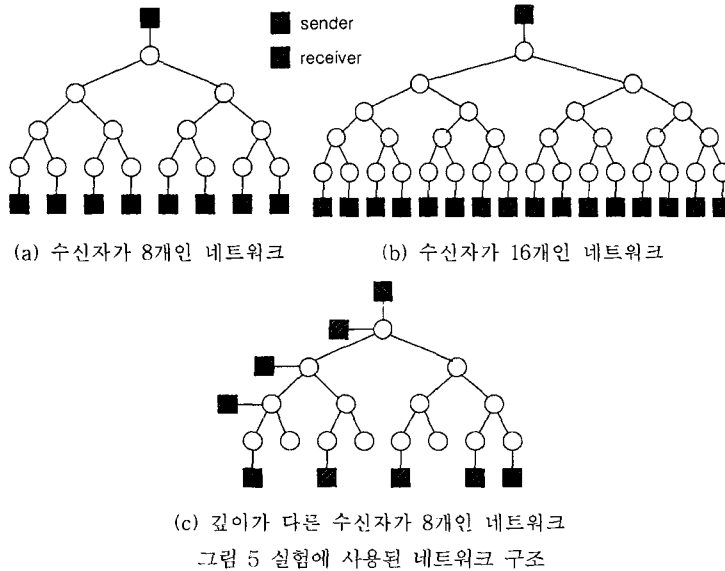


표 1 α, β, γ 값 조합에 대한 응답자 및 수신자의 평균 버퍼 사용량, 응답자의 복구 실패수

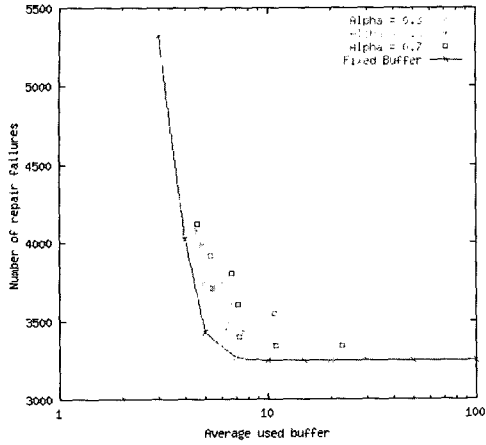
α	β	γ	평균버퍼	실패수	α	β	γ	평균버퍼	실패수	α	β	γ	평균버퍼	실패수
0.3	1.3	0.5	5.975	10427	0.5	1.3	0.5	5.944	10235	0.7	1.3	0.5	6.624	9637
		0.7	7.16	9105			0.7	7.305	8790			0.7	8.496	8262
		0.9	8.188	7630			0.9	8.879	7245			0.9	11.189	6656
	1.5	0.5	6.487	10128		1.5	0.5	7.076	9673		1.5	0.5	9.379	8698
		0.7	7.816	8515			0.7	8.623	8103			0.7	13.593	7436
		0.9	9.363	7498			0.9	10.725	7031			0.9	20.778	6418
	1.7	0.5	7.216	9492		1.7	0.5	8.225	8844		1.7	0.5	22.289	8107
		0.7	8.973	8318			0.7	10.701	7694			0.7	44.631	6929
		0.9	10.142	7229			0.9	13.379	6620			0.9	175.561	6269

수를 나타낸다. α 값이 증가함에 따라 현재의 타이머 값이 영향을 적게 받아 타이머 값의 변화가 더디게 일어나며, β 값이 증가함에 따라 재전송 실패시에 타이머 증가폭이 커지고, γ 값이 증가하면 일정시간동안 재전송 요구가 없을 때 타이머 값을 크게 감소시킴으로써 평균 버퍼 사용량이 늘어나고 그만큼 복구 실패수가 줄어든다. 그림 6의 (d)를 참조하면 이것을 만족하는 α, β, γ 값으로 (0.5, 1.5, 0.9), (0.5, 1.7, 0.9), (0.7, 1.3, 0.9), (0.7, 1.5, 0.9) 등이 있을 수 있다.

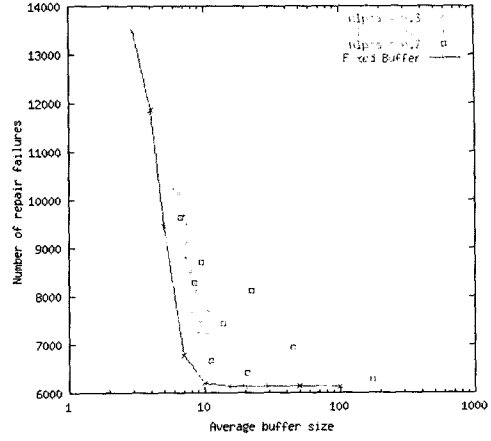
TBM의 확장성을 보이기 위해 고정 버퍼와 TBM을 평균 버퍼 사용량과 복구 실패수 두 가지 요소에 대해 비교한 그림 6에서, 고정버퍼는 실선으로 표시하였고 표 1에서 사용된 TBM의 α, β, γ 값의 각 조합 27개에 대한 점을 함께 나타내었다. 그림 6의 (a), (b), (c)는 16개의 수신자를 갖는 그림 5의 (b) 네트워크에서 연속적인 패킷 손실을 40개당 2, 5, 10개로 하여 한 링크당

패킷 손실이 발생할 확률을 0.4%, 1%, 2.1%로 하였으며, 손실률이 키짐에 따라 고정 버퍼와 TBM 모두 복구 실패수와 평균 버퍼 사용량이 증가한다.

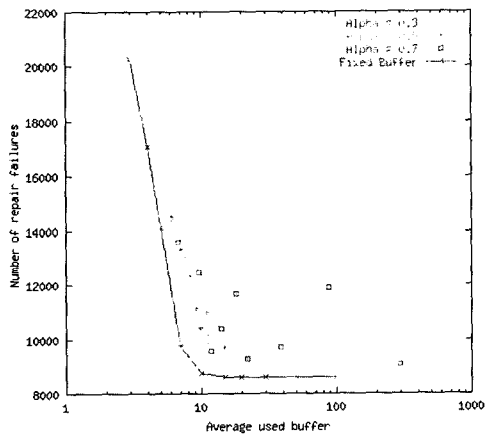
그림 6의 (b)와 (d)에서 연속적인 패킷 손실은 40개당 5개로 동일하지만 (b)는 (d)보다 수신자의 수와 링크 수가 적어져 복구 실패수가 줄어들게 된다. 하지만 지역 복구를 사용하는 LMS를 이용하므로 버퍼 사용량의 변화 추이는 유사한 특성을 갖는다. 즉, 평균 버퍼 사용량을 얼마나 필요로 하는가는 링크나 노드의 수보다는 링크의 패킷 손실률에 많은 영향을 받는다는 것을 알 수 있다. 그림 6의 (d)와 (e)에서는 수신자의 수는 8개로 동일하지만 트리 상에서 수신자의 깊이가 다르게 하여 실험하였으며, 그림 5의 (c)의 네트워크를 이용하였을 경우 수신자들이 송신자에 비교적 가까이 분포하므로 복구를 잘 받을 수 있어 복구 실패수가 줄어들고, 각 점들이 더 밀집하여 분포하지만 전반적인 그래프의 형태



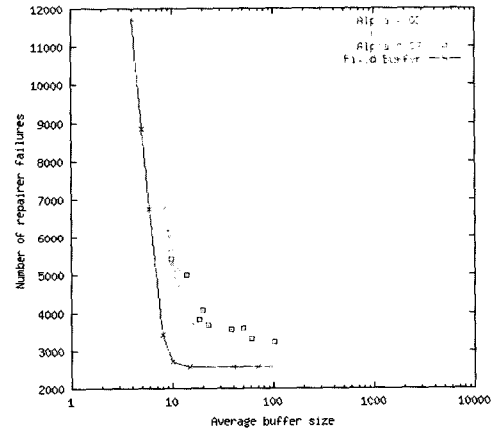
(a) 그림 5의 (b)망, 2/40 errors



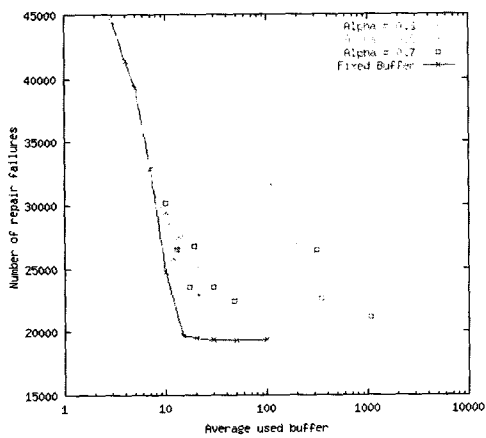
(d) 그림 5의 (a)망, 5/40 errors



(b) 그림 5의 (b)망, 5/40 errors



(e) 그림 5의 (c)망, 5/40 errors



(c) 그림 5의 (b)망, 10/40 errors

그림 6 고정 버퍼와 TBM의 비교

(‘a/b errors’는 b개의 패킷 중 a개를 연속적으로 손실시킨 것을 나타냄)

는 유사하다.

5개의 그림을 살펴보면, TBM이 고정 버퍼보다 좋은 성능을 갖지는 못하지만 손실률이 증가하여 더 많은 버퍼를 필요로 할 때 적절하게 버퍼의 크기를 조절하여 복구 실패수를 줄이는 것을 확인할 수 있다. 실제 네트워크에서는 시간에 따라 멀티캐스트 네트워크의 구성과 각 링크 패킷 손실률이 변화하게 되는 TBM을 이용하면 이러한 상황 변화에 능동적으로 버퍼의 양을 적절히 조절할 수 있다.

기존의 버퍼관리 알고리즘들과 비교하여 본 논문에서 제시하는 타이머를 이용한 버퍼관리 기법은 확장성을

제공하므로 멀티캐스트 트리의 구성이나 링크 오류율에 상관없이 버퍼를 필요한 만큼 할당하여 재전송에 사용할 수 있으며, 실제 상황에서 각 수신자들이 재전송에 필요한 적정량의 고정 버퍼 크기를 미리 예측하고 할당하는 것은 불가능하므로 TBM 기법이 유용하게 쓰일 수 있다. 하지만 TBM은 고정 버퍼에서 불필요하게 확보하고 있는 버퍼 공간을 효율적으로 사용하지 못하며 오류율이 증가함에 따라 고정 버퍼보다 복구 실패수가 더 늘어나게 되는 단점을 지니고 있다. 이러한 문제는 향후 해결해야 할 과제로 남긴다.

5. 결론

멀티캐스트 데이터 전송중 패킷 손실이 발생한 수신자는 재전송을 요청한다. 모든 요청을 송신자가 처리하는 것은 효율적인 방법이 아니므로 올바르게 패킷을 수신한 다른 수신자가 재전송 서버의 역할을 하며, 이를 위해 수신한 패킷의 일부를 저장하고 있어야 한다. 수신자의 버퍼를 어떻게 관리하는가 하는 문제는 수신자의 메모리 관리적인 면뿐만 아니라 손실된 패킷에 대한 재전송 성능에도 영향을 미치는 중요한 요소이다. 본 연구에서 제안한 TBM 기법은 수신자의 패킷 손실 감지 시간과 수신자와 재전송 서버 (응답자) 사이의 패킷 전달 시간에 기초한 타이머를 이용하여 버퍼를 관리한다. 최종적인 타이머 값 결정에 여러 가지 인자들이 작용하며 TBM 기법을 이용하여 네트워크의 오류 상황이나 수신자 수에 영향을 받지 않고 능동적으로 버퍼의 크기를 조절하여 재전송에 사용할 수 있다.

참고 문헌

[1] Deering, S., "Host Extensions for IP Multicasting," RFC1112, January 1989.
 [2] Deering, S. E. and Cheriton, D. R., "Multicast routing in datagram internetworks and extended LANs," ACM Transactions on Computer Systems, pp. 85-110, May 1990.
 [3] Papadopoulos, C. and Parulkar, G., "An Error Control Scheme for Large Scale Multicast Applications," Proc. INFOCOM 1998.
 [4] Guo, K. and Rhee I., "Message Stability Detection for Reliable Multicast," INFOCOM 2000.
 [5] Floyd, S., et al., "A Reliable Multicast Framework for Light weight Sessions and Application Level Framing," Proc. of ACM Sigcomm'95, pp.342-356, September 1995.
 [6] 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.

[7] Costello, A. and McCanne, S., "Search party: Using randomcast for reliable multicast with local recovery," Proceedings of IEEE INFOCOM'99, March 1999.
 [8] Reness, T. and Minsky, Y., et al., "A gossip style failure detection service," Proceedings of Middleware'98, 1998.
 [9] McCanne, S. and S. Floyd, NS (Network simulator), <http://www.nrg.ee.lbl.gov/ns>, 2001.

안상현

정보과학회논문지 : 정보통신
제 30 권 제 1 호 참조



김영민

1999년 서울시립대학교 컴퓨터·통계학과 졸업. 2001년 서울시립대학교 컴퓨터·통계학과 석사. 2002년 3월~현재 서울시립대학교 컴퓨터·통계학과 박사과정. 관심분야는 Wireless Ad-Hoc networks, Wireless routing/forwarding



권영호

2002년 서울시립대학교 컴퓨터·통계학과 졸업. 2002년 3월~현재 서울시립대학교 컴퓨터·통계학과 석사과정. 관심분야는 신뢰적 멀티캐스트, Wireless Ad-Hoc networks