제18회 한국정보처리학회 추계학술발표대회 논문집 제9권 제2호 (2002.11)

# SRM에서 NACK 폭주 억제를 위한 타이머 설정

이연수, 김태환, 김태훈, 박혜련, 이기현
명지대학교 컴퓨터공학과
e-mail : yslee@mju.ac.kr

# Timer Establishment to Suppress NACK Implosion in SRM

Y.S. Lee, T.H. Kim, T.H. Kim, H.R. Park, K.H. Lee
Dept of Computer Engineering, Myeong-Ji University

요 약

SRM[1][2]은 ALF(application level framing)과 LWS(light-weight session)을 위한 신뢰성 있는 멀티캐스트 구조로써 송신자가 아닌 수신자들이 오류가 발생하였는지를 검사하고, 오류가 발생했을 때, NACK 메시지를 전송하여 패킷 손실 복구를 요청한다. 그러나 다수의 수신자들이 송신자에게 패킷의 손실 복구를 요청하게 되면 NACK 폭주(NACK implosion)가 발생한다. 이 논문에서는 NACK 과부하 문제를 해결하기 위해서 수신자가 송신자에서 자신까지의 거리와 request 타이머 인자 값에 의해서 정해지는 구간에서 선택된 임의의 시간동안 기다리는 request 타이머를 개선하여 노드마다 카운터를 주어 안정적인 네트워크에서의 흐름일 경우는 그 구간을 짧게 하고, 반대로 불안정적인 네트워크 일 경우에는 구간을 길게 하여 가변적인 네트워크에 효율적으로 NACK를 억제하는 방법을 제시한다.

## 1. 서론

TCP는 비신뢰적인 IP계층 위에 신뢰성을 추가함으로써, 현재의 대다수 인터넷 응용 프로그램인 WWW, FTP, TELNET 등을 지원하는 전송 프로토콜로 광범위하게 사용되고 있다. 그러나 TCP는 일대일 유니캐스트 통신만을 지원하므로, 화상회의 등의 협동형(collaborative) 응용서비스, 증권 정보 혹은 뉴스 전송 등의 메시지전송(message streaming), 대용량 파일 전송 등의 벌크형 데이터(bulk data) 등의 다수의 사용자 또는 프로세스들이 참여하는 인터넷 응용들을 지원하기에는 네트워크 대역폭이나 호스트 처리율 면에서 비효율적인 한계점을 갖는다.

이러한 새로운 다중 사용자 인터넷 응용들을 지원하기 위해서는 효율적으로 데이터 패킷을 전달하는 IP 멀티캐스트와 신뢰성 있는 전달 서비스를 제공하는 신뢰적 멀티캐스트 프로토콜을 사용하는 것이 효율적이다.

본 연구에서는 SRM에서 발생하는 NACK 과부하 문제를 해결하기 위해 수신자가 오류가 발생하였음을 감지했을 때 NACK 메시지를 즉시 전송하지 않고 우선 송신자에서 자신까지의 거리와 request 타이머 인자 값에 의해서 정해지는 구간에서 선택되는 임의의 시간동안 기다린 후 수신자들로부터 중복된 NACK 메시지를 받지 않으면, 비로소 NACK 메시지를 전송하는 기존의 random한 타이머 설정 방법을 개선한 동적인 타이머 설정 방법을 제시한다.

2장에서는 멀티캐스트 신뢰성 있는 멀티캐스트 전송을 위해 제안된 3가지 프로토콜 클래스인 트리기반 ACK 방식, NACK 방식, FEC방식을 소개하고, 3장에서는 SRM에 대하여 살펴본 후 타이머 설정 방법을 설명한다.

4장에서는 제안하는 동적인 타이머 설정 방법에 대해 설명한다. 끝으로 5장에서는 분석을 바탕으로 결론을 내리고 앞으로의 과제에 대하여 살펴본다.

## 2. 관련연구

멀티캐스트 신뢰성 있는 멀티캐스트 전송을 위해 제안된 3가지 프로토콜에는 트리기반 ACK 방식, NACK 방식, FEC방식등이 있다.

2.1 트리기반 ACK 방식

트리기반 ACK 멀티캐스트 방식은 멀티캐스트 사용자를 연결하는 수송계층의 논리적 트리를 구성하여 트리 계층 구조를 이용하여 오류를 복구하는 방식이다. 트리를 통해 각 수신자들은 부모-자식(parent-children) 관계를 형성할 수 있으며, 각 부모 노드들은 자식 노드들에 대한 오류 복구 기능을 제공한다. 부모 노드가 재전송을 요구하는 패킷을 가지고 있지 않은 경우에는, 상위 부모노드에게 재전송을 요청하여 결국 데이터 송신자에게 재전송 요구가 전달될 수 있다.

부모노드는 각 자식노드로부터 ACK 패킷 및 손실정보를 받은 뒤 오류를 복구하고, 자기 서브 그룹에 대한 데이터 송수신 상태 정보를 상위, 차상위 부모 노드에게 전달한다. 이러한 상태정보에 입각하여 송신자는 흐름 및 혼잡 제어를 수행하게 된다. 트리기반 오류제어에서는 이처럼 수많은 수신자들을 하나의 트리 구조로 연결하여 확장성 문제를 해결하고자 한다. 하지만 어떻게 전체 그룹을 하나의 트리로 구성할 것인지에 대한 만족할 만한 해법이 나오지 않았으며, 특히 어떤 수신자 혹은 개체를 부모 노드로 정할 것인지 등의 이슈가 남아 있다. 또한 망의 혼잡제어를 위해 각 수신자의 정보를 송신자에게 보낼 필요가 있는 경우, 해당 제어 패킷이 트리를 따라 송신자에게 전달되는 동안 일정시간의 지연(dealy)이 발생한다는 문제점도 있다. 대표적인 프로토콜로써는 TMTP(Tree-Based Multicast Transport Protocol)[8] 및 TBRM(Tree Based Reliable Multicast)[3] 등이 있다.

2.2 NACK 기반 멀티캐스트 방식

NACK 기반 멀티캐스트는 오류 재전송을 요구하는 NACK 패킷을 전체 그룹에 전송하는 방식이다. 가까이에 있는 성공적인 수신자가 있을 경우, 이러한 NACK 패킷에 응답한다. 이 경우 NACK 패킷을 여러 수신자들이 동시에 발생시켜 NACK 패킷이 폭주할 우려가 있으므로, 각 수신자는 타이머(timer)를 이용하여 적절한 시간동안 다른 NACK 패킷이 이미 발생되었는지를 파악한다. 이러한 기법을 NACK 억제(suppression) 기법이라 한다. NACK 기반 오류제어에서는 오류복구 기능을 송신자가 아닌 가까이에 있는 수신자의 도움으로 해결하여 확장성을 높이고자 한다. 이 방식에서는 특히 NACK 억제를 위해 사용되는 타이머의 동작이 전체 성능에 영향을 주며, 특히 모든 수신자에게 멀티캐스트 송신 능력을 요구한다. 또한 ACK 패킷의 기능중인 하나인 송신버퍼의 방출(release 혹은 flush) 기능이 없다는 문제점을 지니고 있다. 이 방식의 대표적인 프로토콜로서는 SRM(Scalable Relialble)을 들 수 있다.

2.3 FEC 기반 멀티캐스트 방식

ACK/NACK 기반 복구 오류복구에서는 수신자의 패킷 수신정보를 송신자에게 전달하여 송신자가 패킷을 재전송하는 ARQ(Automatic Request)방식인 반면에, FEC방식에서는 송신자가 데이터 송신단계에서부터 parity bits등의 redundancy를 부과하여 데이터를 전송한다. 즉, 수신자로부터 피드백(feedback) 없이, 패킷 손실이 발생했을 경우 다른 패킷의 redundancy 정보를 이용하여 손실된 패킷을 복구하는 방식이다. 이러한 기법을 FEC 방식이라 하며, 최근에 큰 주목을 받고 있다. 특히 위성 등의 비동기(asynchronous) 네트워크에서 적용이 용이하며, 재전송 및 오류복구 등으로 인한 추가 지연시간이 소요되지 않는다. 이를 위해 별도의 FEC 코딩(coding) 방식이 요구되며, 안정적인 망에서 사용하기에는 불필요한 오버헤드(Overhead)가 발생할 수 있다. 이 방식의 대표적인 프로토콜로써는 RMDP (Reliable Multicast Dissemination Protocol)[3]가 있다.

## 3. SRM(Scalable Reliable Multicast)

SRM은 공유 화이트보드(shard whiteboard)와 다자가 응용을 지원하기 위해 개발된 것으로 실제 MBone에서 실험되고 있다. SRM은 수신자 주도방식 바탕 위에 송신자 뿐 아니라 다른 수신자들도 손실 복구를 수행하는 것을 허용한다. 이를 위해 NACK은 세션 전체로 멀티캐스트되어야 하는데 이때 NACK implosion 현상을 방지하기 위해 NACK 억제기법을 사용한다. 복구 시에도 동일한 패킷이 중복되는 것을 막기 위해 복구 타이머(repair timer)를 이용하여 얼마간 기다렸다 멀티캐스트하는 억제 방법을 사용한다.

타이머 값의 설정이 중요한 요소인데, 요청 타이

머의 경우 $2^i$[C1*d, (C1+C2)*d] 구간에서 임의의 (random) 값을 선택하도록 한다. d는 수신자 자신과 송신자와의 추정된 RTT값으로, 거리를 쓰는 이유는 보통 송신자와 가까운 수신자가 먼저 손실을 감지할 가능성이 크므로, 가까운 수신자의 NACK 전송을 장려하고, 거리가 먼 수신자는 그 만큼 오래 기다리므로, 그 동안 NACK이 더 가까운 수신자에 의해 이미 전송된 것을 알 수 있어 NACK을 억제할 수 있기 때문이다. 임의의 값을 쓰는 것은 거리가 같은 수신자들간에 발생 가능한 중복 NACK을 줄이자는 목적이다. 비슷한 식으로 복구 타이머도 랜덤한 값을 선택한다.

초기 LAN을 위한 NACK 억제 기법과 달리 SRM은 세션 메시지 교환에 의해 추정된 지연시간 값에 기반하여 타이머를 적응시켜 WAN에서의 성능을 향상시키고자 하였다. 그러나 타이머를 설정하기 위해 지연시간을 추정하는 것은 세심한 주의를 요한다. 특히 세션 멤버쉽이 동적인 상황과 가변적인 네트워크 상황에서 최적의 타이머 값 설정이 어렵다.

### 3.1 NACK 타이머

송신한 데이터의 손실이 발견되면 NACK 타이머가 작동하기 시작한다. 타이머가 종료되는 시간은 다음의 범위 안에서 구한다.

$$2^i*[C1*d_{SA}, (C1+C2)*d_{SA}](sec)$$

$d_{SA}$는 호스트 A에서 손실된 데이터의 원래 소스 S까지의 한 방향 거리이고 C1과 C2는 요청 알고리즘에서의 인수들인데 초기값은 2로 준다. 또 i는 타이머를 철회한 수이고 초기값은 0으로 준다.

타이머가 종료되면 호스트 A는 손실된 데이터에 대한 NACK을 멀티캐스트하고 복구 데이터를 기다리기 위해 타이머의 시간을 두 배로 설정한다.

만약 호스트 A가 손실된 데이터에 대한 자신의 타이머가 끝나기 전에 NACK을 받는다면 호스트 A는 타이머를 다음의 범위로 고친다.

$$2^{(i+1)}*[C1*d_{SA}, (C1+C2)*d_{SA}](sec)$$

### 3.2 복구 타이머

호스트 B가 A로부터 손실된 데이터의 요청을 받고 그 데이터를 가지고 있을 때 B는 복구 타이머를 다음의 범위로 설정한다.

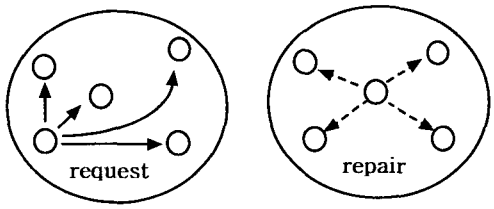
$$[D1*d_{AB}, (D1+D2)*d_{AB}](sec)$$

$d_{AB}$는 호스트 B의 A까지의 한 방향 거리이고 D1, D2는 초기값이 $\log_{10}(G)$로 정해지는 복구 알고리즘의 인수들이다(G는 현재 세션의 참가자 수).

호스트 B가 타이머가 종료되기 전에 손실된 데이터의 복구 데이터를 받는다면 B는 타이머를 종료시킬 것이고 그렇지 않다면 타이머가 종료되었을 때 복구 데이터를 멀티캐스트 시킨다.

### 3.3 호스트간의 거리 계산

위의 타이머 범위에서의 거리 d는 호스트 A에서부터 호스트 B까지 패킷이 전송되는 시간을 말한다. 이것은 각 참가자가 전송하는 타임스탬프를 사용하여 계산할 수 있다. 예를 들어 호스트 A가 패킷 P1을 T1 시간에 전송하였고 호스트 B는 T2 시간에 P1을 받았다고 가정하면 얼마 후에 T3에 호스트 B가 만드는 P2에는 (T1, d)가 포함되어 있다. 이 때 d는 T3-T2이다. 호스트 A가 T4에 P2를 받았다면 호스트 B와 호스트 A 사이의 거리는 (T4-T1-d)/2로 예측되어질 수 있다. 주의할 것은 이러한 계산은 경로가 양 방향으로 대칭적이라고 가정할 때이다.




(a)손실 복구 요청 (b)손실 복구 패킷 전송

**[그림1] SRM에서의 손실 복구 요청 및 복구 패킷 전송**

## 4. 제안방법

기존의 SRM에서는 NACK-과부하의 문제를 해결하기 위해서 각 수신자들이 서로 다른 제어 메시지의 전송 시점을 가지도록 조정한다. 각 수신자에서의 전송 시점을 결정하기 위해서 SRM은 랜덤한

request 타이머 파라미터 값을 사용한다. 그러나 제안하는 타이머 설정 방법은 이러한 파라미터 값을 동적으로 조정할 수 있도록 하였다.

제안하는 타이머 설정 방법은 각 노드마다 카운터(K)를 두어 네트워크의 변화에 적절히 대응할 수 있도록 타이머 파라미터를 값을 설정하는 방법이다.

네트웍이 안정적일 경우 NACK 메시지가 중복될 경우는 적기 때문에 request 파라미터 값을 작게 설정하도록 하고, 네트웍이 불안정적일 경우에는 request가 많아져 NACK 메시지가 중복될 수 있기 때문에 request 파라미터 값을 높게 주어 NACK 메시지 중복을 피할 수 있게 된다. K값 변화에 대한 알고리즘은 그림2와 같다.

```
K=1
if 네트웍에 오류 발생시
  K=K+1
  2^i*[C1*dSA, (C1+(K*C2)*dSA]
else 네트웍이 안정적 일 때
  K= 1/2 k
  2^i*[C1*dSA, (C1+(K*C2)*dSA]
else
  K=1
```

**[그림2] 카운터에 대한 알고리즘**

Request 타이머 파라미터는 카운터 값의 변화로 인해 가변적인 네트워크에서 중복되는 NACK 메시지의 수가 감소되는 것을 알 수 있다.

## 5. 결론 및 향후 연구방향

멀티캐스트 신뢰성 있는 멀티캐스트 전송을 위해 제안된 3가지 프로토콜 클래스인 트리기반 ACK 방식, NACK 방식, FEC방식을 소개하였고, SRM에 대하여 살펴본 후, 제안하는 동적인 타이머 설정 방법을 설명하였다.

기존의 SRM에서는 가변적인 네트워크에 적합하지 않는 랜덤한 타이머 방식으로 NACK implosion이 발생하였지만 제안하는 카운터를 구성하는 방식으로 동적으로 타이머 값을 설정할 수 있어 NACK implosion을 막을 수 있다.

향후에는 가변적인 네트워크에서 repair 타이머 파라미터의 값을 동적으로 조정하는 부분을 좀 더 보완하려고 한다.

## 참고문헌


[1] Sally Floyd, Van Jacobson, Steven McCanne, "A Reliable Multicast Framework for Light-Weight Session and Application Level Framing" ACM SIGCOMM 95, pp 342-356, August 1995

[2] Sally Floyd, Van Jacobson, Ching-Gung Liu, Steven McCanne, and Lixia Zhang, "A Reliable Multicast Framework for Light-Weight Session and Application Level Framing". IEEE/ACM Transaction on Netowrking November. 1996.

[3] B. Whetton, et al., "The Reliable Multicast Transport Protocol," IETF Internet Draft, draft-whetton-rmtp-ii-00.txt, Apr. 1998.

[4] B. Adamson, C. Borman, S. Floyd, M. Handly, J. Marker, "NACK-Oriented Reliable Multicat(NORM) Protocol Building Blocks," IETF Internet Draft, draft-ietf-rmt-norm-bb-00.txt, Mar. 2000

[5] S. McCanne, V, Jacobson, "Receiver-driven Layered Multicast" ACM SIGCOMM'96. Aug. 1996

[6] A. Koifman and S. Zabele, "RAMP: A Reliable Adaptive Multicast Protocol", IEEE INFOCOM '96, March, 1996

[7] R. Aliello, E. Pagani and G. P. Rossi, "Design of a Reliable Multicast Protocol", IEEE INFOCOM '96, pp74-81, March 1992.

[8] Yavatkar, Griffioen, and Sudan, A Reliable Dissemination Protocol for Interactive Collaborative Applications, ACM Multimedia 1995, San Francisco, CA, November 1995.