 멀티캐스트 전송을 위한 오류제어기법의 분류
Classification of Reliable Multicast Transport Protocols

고석주(S.J. Koh)  선행표준연구팀 선임연구원
김용운(Y.W. Kim)  선행표준연구팀 선임연구원
박정수(J.S. Park)  선행표준연구팀 연구원
김현수(G.S. Kim)  선행표준연구팀 선임연구원
김용진(Y.J. Kim)  선행표준연구팀 선임연구원, 책임

멀티캐스트 전송을 위한 오류제어기법에 대하여 많은 연구가 진행되어 왔다. 멀티캐스트 전송의 주요 고려사항은 신뢰성 및 확장성이다. 본 논문에서는 이 두 가지 문제를 해결하기 위해 제안되어 온 연구 결과를, 특히 오류제어기법 측면에서 정리 및 분류하는 데 초점을 둔다. 이러한 분류를 통해 향후 수송계층 프로토콜의 오류제어기법을 개발함에 있어서 고려해야 할 사항을 점검해 볼 수 있다.

I. 서론

신뢰성 프로토콜이란 통신 사용자에게 오류없는(error-free) 데이터 전송을 제공해주는 프로토콜을 말한다. 수송계층에서의 신뢰성은 주로 전송품질 저하로 인한 데이터 오류를 복구하는 오류제어기법에 의해 제공된다.

TCP는 중단간의 통신 사용자 사이에 신뢰성을 제공하는 수송계층 프로토콜이다. TCP는 현재 유일한 신뢰성을 보장하는 유니캐스트 프로토콜로 알려져 있다. 사실상, 향후에도 인터넷 상의 유니캐스트 통신의 신뢰성을 보장할 때로 대부분의 사용자는 TCP를 사용할 것으로 예상된다.

TCP의 오류제어방식은 ACK(acknowledge-
ment) 메시지를 이용한다. 성공적으로 수신된 테이터에 대하여 수신자는 ACK 메시지를 보낸다. ACK 메시지는 태게 다음 번에 받아야 할 패킷 번호를 포함하며, 최근 메시지의 성공적인 수신을 의미한다. 패킷 번호 누락 등의 데이터 손실이 있을 때, TCP는 오류복구를 위한 재전송(retransmission) 모드로 들어간다.

TCP는 일종의 송신자주도(sender-initiated) 오류제어방식이다. 즉, 데이터 전송 오류의 탐지 및 재전송 역할을 송신자가 책임진다. 하지만, TCP 오류제어방식은 멀티캐스트 환경에 그대로 적용하기에는 무리가 따른다. 즉, 송신자는 전송한 패킷에 대하여 각 수신자로부터 ACK 메시지를 받어야 하며, 이러한 메시지의 양은 수신자의 수가
많아질수록 급격히 증가한다.

이러한 문제를 ACK 폭주(implosion) 문제라 부르며, 송신자의 수가 많아질수록 문제는 더욱 심각하다. 즉, 송신자주도방식은 확장성 scalability 문제를 발생시킨다. 이러한 확장성 문제를 해결하기 위해 지난 10여 년 간 많은 대안들이 제안되어 왔다. 하지만, TCP와는 달리, 멀티캐스트 환경에서는 “단 하나의 보편적인(one-size-fits-all) 방법”은 개발되기 어려울 것으로 여겨진다. 신뢰성에 대하여 각 멀티캐스트 옵션은 상이한 요구사항을 가지므로, 현재 상용화된 응용서비스만 고려할 뿐더러, 적어도 몇 종류의 프로토콜은 있어야 할 것이다. 또한 아주 고유한 요구사항을 갖는 응용서비스는 별도의 신뢰성 프로토콜을 필요로 할지 모르다. 본 논문에서는 확장성 문제를 해결하기 위해 지금까지 제안되어 온 오류제어기법들을 각 특징별로 정리 및 분류하고자 한다.

서문에 이어 제II장에서는 먼저 기존에 제안된 오류제어기법들을 분류 및 정리한다. 제III장에서는 제II장에서 분류된 방식 중에 가장 대표적인 프로토콜들을 중심적으로 살펴본다. 제IV장에서는 이러한 분류 및 정리를 토대로, 향후에 수송계획 프로토콜의 오류제어기법 개발시 고려해야 할 점을 살펴본다.

II. 오류제어기법의 분류

본 장에서는 현재까지 제안되어 온 신뢰성 프로토콜들의 오류제어기법들을 각 특징별로 분류한다. 관점에 따라 여러 가지 방법으로 분류할 수 있었으나[1, 2], 본 글에서는 특히 송신자주도 오류제어기법의 확장성 문제를 해결하기 위해 어떠한 방식이 사용되었는가에 초점을 두고 기존 방식들을 다음과 같이 5가지로 분류한다.

- 수신자 주도(receiver-initiated)
- 토큰 기반(token-based)
- 서버기반(server-based)로컬복구(local recovery)
- 트리 기반(tree-based)로컬복구(local recovery)
- 응용서비스 기반(application-specific)

1. 수신자주도 오류제어기법

송신자주도기법에서는 송신자 측에서 오류제어를 책임지는 것과는 달리, 수신자주도기법은 개별 수신자가 오류제어의 책임을 진다. 송신자는 수신자로부터 NAK(Negative ACK)가 올 때까지 데이터 전송을 계속한다. 수신자로부터 NAK 메시지가 오는 경우, 송신자는 해당 데이터를 특정 수신자에게 재전송한다.

각 수신자는 데이터 오류를 탐지한 후, NAK 메시지를 송신자에게 유니캐스트로 전송한다. 이 경우에 여러 수신자가 같은 NAK 메시지를 하나의 수신자에게 전송하는 문제가 발생할 수 있다. 이를 NAK 폭주라 한다. 이를 방지하기 위해 NAK 메시지를 그룹 전체에게 멀티캐스트 하는 경우도 있다. NAK 메시지를 전체 그룹에게 멀티캐스트 하는 경우, 각 수신자는 일정 시간을 기다렸다가 NAK 메시지를 전송한다. 이는 다른 수신자가 이 미 같은 NAK를 전송했을 수도 있기 때문이다. 이러한 방식으로 NAK 발생을 억제(suppression)하는 메커니즘을 “slotting & damping” 방식이라 부른다.

NAK 메시지를 멀티캐스트 하는 경우, 이는 다른 수신자에게 더 이상의 같은 NAK를 보내지 말하는 의미를 갖기도 하며, 또한 주의의 수신자 중에서 해당 데이터를 성공적으로 수신한 수신자가 있는 경우에는 자신에게 데이터 재전송을 요구하

하지만 NAK 방식 자체로는 흐름제어(flow control) 측면에서 완전한 수송계층 프로토콜이 아니다. 송신자 입장에서 윈도 크기(window size)를 제어하기 위해서는, 어떤 형태로든 ACK 정보를 받아야 하는데, 이것이 수신자로부터 ACK 메시지에 상응하는 정보가 도착해야 수신자가 메모리를 방출(release)할 수 있다. 이를 위해서 수신자에서 송신자에게 주기적으로 데이터 수신 성공을 알리기 위해 주기적 폴링(periodic polling) 방식을 사용하기도 한다.

이처럼 수신자주도 오류제어방식에서는 모든 수신자가 오류제어의 책임을 진다. 이에 비해 제2절에 기술된 토론 기반 오류제어방식에서는, 토론을 가진 사용자만이 오류제어를 담당한다.

3. 서버기반 로컬복구기법

위의 방식에서는 송신자와 수신자가 직접적으로 오류제어에 참여했던 것과는 달리, 서버기반 로컬복구방식에서는, 미리 설정된 하나 혹은 여러 개의 서버가 오류제어 및 데이터 재전송을 담당한다. 따라서 송신자의 처리부담을 여러 개의 서버로 분산시킬 수 있다. 이러한 방식을 사용하는 프로토콜은 LBRM(Log-Based Reliable Multicast) 등이 있다[7].

서버기반 오류제어방식에서 각 서버는 송신자의 모든 데이터를 기록하고 있다. 서버는 수신자와 데이터 수신에 대해 송신자에게 ACK 메시지를 보낸다. 수신자는 데이터 손실에 대해 송신자와 아닌 서버에게 NAK 메시지를 보낸다. NAK 메시지를 받은 서버는 서버를 저장해 놓은 데이터를 이용하여 해당 수신자에게 데이터를 재전송한다.

이 방식은 여러 개의 서버를 두어 트래픽 부하를 분산시키는 장점이 있으나, 별도의 서버 선정
및 관리, 그리고 추가적인 메모리 할당 등의 오버헤드를 요구한다.

4. 트리 기반 로컬복구기법


트리는 여러 개의 서브트리로 구성되며, 각 서브트리의 루트(root)가 오류 복구를 제어한다. 각 서브트리 안의 child는 또 다른 하위 서브트리의 루트가 될 수 있다. 이러한 제어용 트리의 구성은 새로운 멤버의 추가 혹은 기존 멤버의 탈퇴 등으로 변경될 수 있다. 새로운 멤버는 “expanding ring search”(TTL 값을 확장해나가며 가장 가까운 루트를 탐색하는 방법) 기법을 이용하여 제어용 트리에 가입한다.

각 child는 정공격의 데이터 수신의 경우 루트 혹은 parent에게 ACK 메시지를 보낸다. 이러한 측면에서 제어용 트리를 ACK 트리라고도 부른 다. 각 서브트리의 루트는 자신이 관할하는 child 들의 데이터 수신상태를 취합하여 상위 루트에게 주기적으로 전달한다.

데이터 손실의 경우 수신자는 자신의 parent에게 NAK을 유니캐스트로 알리거나(RMTMP의 경우) 혹은 서브 트리 내에 멀티캐스트로 전달한다(TMTMP의 경우). 데이터 재전송의 경우, 수신된 NAK 메시지 수에 따라 유니캐스트 혹은 멀티캐스트 방식으로 수행된다.

트리 기반 오류제어기법은 매우 좋은 확장성을 갖는다[2]. 하지만 네트워크 계층의 트리와는 별도의 제어용 트리를 구성 및 관리해야 하는 오버헤드가 요구된다. 또한 단일 세션이라 하더라도 송신자가 여러 명 있는 경우, 여러 개의 트리가 구성되어야 한다. 한편, 최근에 제안된 LoraX[10] 프로토콜에서는 단일 멀티캐스트 세션에서 여러 송신자가 존재하는 “many-to-many” 서비스를 위해 공유형(shared) ACK 트리방식을 제안하였다.

5. Application-Specific 오류제어기법

모든 응용서비스에 적합한 오류제어기법을 개발하기는 매우 어려운 일이다. 실제 최근 연구동향을 살펴보면, 특정 응용서비스 혹은 특정 목적에 적합한 오류제어기법을 개발하고자 하는 노력이 찾아볼 수 있다. 이에 해당하는 예로는 MFTP (Multicast File Transfer Protocol)[11], STORM (Structure-Oriented Resilient Multicast)[12], PFGM (Pretty Good Multicast 및 Pragmatic General Multicast)[13] 프로토콜, 그리고 ALF(Application Level Framing)[14] 접근방식 등이다.

MFTP는 StarBurst사에 의해 개발되고 있으며, 대용량 파일전송서비스에 적합한 application-specific 멀티캐스트 프로토콜이다. 데이터 전송은 “pass” 단위로 이루어진다. 첫번째 pass에서 전체 파일을 그룹 사용자에게 전달된다. 파일 데이터를 성공적으로 수신한 멤버는 송신자에게 ACK를 전송하며, 데이터 손실을 경험한 수신자는 NAK 메시지를 전송한다. 두번째 pass에서, 송신자는 손실된 데이터에 대하여 파일 전송을 수행한다. 이와 같은 방식으로 모든 수신자가 모든 데이터를 성공적으로 수
신한 때까지 테이터 전송이 반복된다.

STORM은 연속적인 메시지를 위한 프로토콜이다. 따라서 지연(delay)에 매우 민감한 서비스를 목표로 개발되었다. 전체 수신자들은 동적인(dynamic) 제어그래프(control graph)로 구성된다. 트리기반방식과는 달리, 한 수신자가 여러 개의 parents를 지정하여 parent 노드들의 목록을 관리한다.

데이터 손실의 경우, 수신자는 목록에 있는 parent 노드들에게 차례대로 NAK 메시지를 전송한다. 첫번째 parent에게 응답이 없을 경우 그 다음 parent에게 NAK를 전송한다. 즉, 번거리에 위치한 송신자에게 직접적으로 제전송을 요청하는 대신에, 가까이 있는 여러 parents에게 데이터 제전송을 요청하는 방법이다. 따라서 그만큼의 제전송 지연을 단축시킬 수 있다.

PGM은 가장 최근에 제안된 프로토콜이다. 구현이 매우 간단하며 확장성이 매우 좋다. 트리기반방식이 제어용 트리를 구성하여 오류정보를 수행한 것과는 달리, 네트워크 계층의 멀티캐스트 트리를 직접 오류 정보에 사용한다. 따라서, 현행 인터넷상에서 곧바로 구현될 수 있는 방식이다. PGM 프로토콜에 대한 상세한 기술은 다음 장에서 이루어진다.

ALF(Application Level Framing)[14] 방식에서는 다양한 응용서비스의 신뢰성을 수송계층 프로토콜에서 제공하기 어려기 때문에, 각 응용 서비스 특성 및 요구사항에 맞도록 응용계층에서 해결하고자 하는 접근방식을 취하고 있다. ALF에서는 모든 데이터가 "application data" 단위로 교환된다. 각 데이터는 application-specific name space로 표현된다. 또한 각 응용 데이터는 신뢰성 요구에 따라 부호화(encoding)된다. 즉, 수송계층에서는 종단간의 데이터 전송만을 담당하고, 신뢰성 제공은 응용계층에서 담당하고자 하는 방식이다.

### III. 주요 프로토콜의 오류제어기법

본 장에서는 지금까지 논의되어 온 오류제어기 법 분류를 토대로, 각 분류별로 가장 대표적인 프로토콜에 대하여 보다 상세히 살펴보고자 한다.

#### 1. SRM

SRM(Scalable Reliable Multicast)[3]은 설계현재의 MBONE 상에서 실행되고 있으며, GlobalCast 사(www.gcast.com)에 의해 개발되고 있다. 실제 MBONE의 whiteboard(wb) tool은 현재 SRM 프로토콜을 사용하고 있다. SRM의 목표는 다른 프로토콜과 마찬가지로 확장성이다. 또한 기존의 TCP 동작원리를 그대로 사용하고자 노력한다.

SRM은 수신자주도 오류제어방식을 사용한다. SRM에서 그룹 멤버들을, 다른 수신자가 데이터 손실을 경험할 때에, 오류 복구를 위해 서로 협력한다. 또한 그룹 멤버간의 상호 협력을 통해 제어 메시지의 폭을 방지하고자 한다.

SRM은 오류 복구를 위해 단지 멀티캐스트 전송만을 사용한다. 그만큼 구현하기에 간단하다. SRM은 세션 메시지 교환을 통해 다른 멤버들과의 거리를 측정한다. 수신자가 데이터 손실을 탐지하면, NAK 메시지를 멀티캐스트로 전송하여 도움을 요청한다. 근처에 위치한 다른 수신자는 손실 수신자에게 도움을 줄 수 있다. 오류복구 요 청 메시지의 범람을 방지하기 위해, 각 손실 수신 자는 일정시간 동안 기다린 후 요청 메시지를 전송한다. 사실상 이러한 대기시간(waiting time)은 송신자와의 거리에 비례하게 확정된다. 대기시간
동안에 다른 손실 수신자가 이미 요청 메시지를 보낸 경우 자신은 NAK 메시지를 전송하지 않는 다. 이러한 메커니즘으로 요청 메시지를 억제하게 된다. SMR 구현을 위해 모든 수신자는 최근에 받은 일정 양의 데이터를 저장하고 있어야 한다.

데이터 손실에 상응하는 데이터 재전송도 NAK 전송과 비슷한 메커니즘으로 이루어진다. 데이터 수신에 성공한 수신자가 손실 수신자에게 데이터 를 재전송하는 경우에도 일정기간 기다린 다음 멀티캐스트 방식으로 데이터를 재전송한다. 만약 이미 다른 성공 수신자가 재전송을 한 사실을 알게 되면, 자신은 재전송을 하지 않는다. 이를 ‘재전송 억제’라 부른다.

모든 메시지가 항상 멀티캐스트로 전송되는 점은 그룹 멤버들이 별도의 대역을 저장 없이도 마음대로 할 수 있는 수단을 제공한다. 또한 적절한 대기시간 변수의 설정을 통해 중복되는 재전 송 요청 및 재전송 트래픽을 억제할 수 있다. 하지만 모든 메시지를 멀티캐스트로 전송하기 때문 에, 제어 메시지의 패킷이 수명도 그만큼 높다.

2. RMP

RMP( Reliable Multicast Protocol)[6]는 West Virginia 대학과 NASA가 연합하여 개발하는 중이며, SRM처럼 GlobalCast에 의하여 사용되는 종이다. RMP는 토론 기반 오류예방 메커니즘을 사용한다. 대개 수신자가 그룹이 하나의 링 형태로 구성되어 토론을 순환시킨다. 토론을 가진 수신 자가 전체 그룹에 대한 최근 패킷들을 제어한다.

성공적인 데이터 수신의 경우 토론 사용자만 송신자에게 ACK 메시지를 보낸다. 토론 사용자 는 자신이 필요로 하는 모든 데이터를 성공적으로 수신한 다음, ACK 메시지를 보내고, 토론을 다 음 사용자에게 전달한다. 송신자는 ACK를 받을 때마다 왼도 크기를 증가시킨다.

데이터 손실의 경우, 손실 수신자는 NAK 메시 지를 전체 그룹에 멀티캐스트 방식으로 전송한다. RMP의 NAK 메시지는 단지 송신자의 왼도 크기를 감소시키는 데 사용된다. 나중에 손실 사용자 가 토론을 가게 되었을 때, 그동안 손실되었던 데이터를 송신자로부터 전달받는다.

RMP는 이처럼 토론을 사용하여 그룹 사용자의 트래픽을 제어한다. 하지만 데이터 손실이 발생한 시점에서 즉시(timely) 재전송하는 것이 아니 라, 토론을 가진 사용자만이 오류 복구를 하게 되므로, 실시간 및 대화형 서비스에는 적용하기 어렵다.

3. LBRM

LBRM( Log-Based Receiver-reliable Multicast)[7] 프로토콜은 서버 기반 오류제어를 수행 한다. 여러 개의 logging 서버가 선정되며, 각 서버는 송신자가 보내는 데이터를 저장해놓는다. 서버 들끼리 계층구조로 구성될 수도 있다. 서버는 성공적인 데이터 수신에 대하여 송신자에게 ACK 메시지를 전송한다.

한편 수신자는 데이터 오류를 탐지하는 경우, 송신자와 직접 연락을 취하는 대신에, 가까이에 위치하는 서버에게 NAK 메시지를 전송한다. 서 버는 해당하는 데이터를 송신 수신자에게 유니캐스트 방식으로 재전송하느니, 아니면 로컬 그룹 에게 멀티캐스트로 전송한다. 보통 로컬 그룹 내 의 NAK 송신자가 많은 경우 멀티캐스트로, 반대 의 경우에 유니캐스트로 데이터를 재전송한다.

이 방식은 전체 수신자 그룹을 로컬 그룹으로 조직화하여, 제어 메시지의 부하를 극히 분산시키는 효과를 주지만, 별도의 서버를 관리 유지하고 또한 서버에 추가적인 메모리 할당이 요구
4. RMTP

RMTP[9]는 SRM, RMP와 함께 GlobalCast사의 연구진에 의해 개발되는 중이다. RMTP는 MBONE tunnel 환경에서 수 차례 실험되어 왔으며, 'native multicast(non-tunnel)' 환경에서도 실험되고 있다. RMTP는 TMTP와 함께 트리 기반 오류제어를 수행하며, 늦은 확장성을 갖는 것으로 알려져 있다.

RMTP는 전체 수신자 그룹을 트리 형태로 구성한다. 이러한 별도의 트리를 제어용 트리라 한다. 제어용 트리는 여러 개의 서브 트리로 구성된다. 이러한 트리는 'expanding ring search' 기법을 이용하여 구성된다.

RMTP는 각 영역을 n-level의 계층구조로 나누어 신뢰성을 제공한다. 각 영역은 DR(Designated Receiver)를 선출한다. DR은 소속 영역의 수신자들 이 어떤 패킷을 성공적으로 수신하였는지의 여부에 대한 상태 정보를 수집하여 상위 DR에게 전달한다.

데이터 손실의 경우, 만약 DR이 손실 데이터를 보유하고 있다면, 그대로로 로컬 영역에 재전송한다. 그렇지 않은 경우, 상위 DR에게 재전송 요청 메시지를 보낸다. 이러한 절차들이, 모든 수신자가 특정 window의 모든 데이터를 수신할 때까지, 각 영역별로 수행된다.

RMTP의 고유 특징은 계층적 구조를 사용한다는 점과, 상대 메시지를 영역별로 취합한다는 점이다. 이러한 특징으로 인해 송신자에게 상대(status) 메시지가 집중되는 현상을 방지할 수 있다. 또한 그룹 전반에 미치는 메시지의 수를 최소화할 수 있다.

5. PGM


PGM의 가장 큰 특징은 단순성과 뛰어난 확장성에 있다. PGM은 특히 실시간 응용서비스에 적합한 방식이다. 또한 기존의 트리 기반 오류제어 방식과는 달리, 네트워크 계층의 멀티캐스트 트리를 그대로 수송계층의 오류제어에 사용한다. 한 가지 주목할 것은 PGM의 트리는, TMTP 혹은 RMTP에서 사용되는 로컬그룹 제어를 위한 논리적 트리가 아니라, 실제 멀티캐스트 전송 트리 그 자체라는 점이다. 환면 대부분의 reliable 멀티캐스트 프로토콜이 UDP 위에서 동작하는 것과는 달리, PGM은 TCP, UDP 모두 IP 위에서 동작하도록 설계되었다. 따라서 현재 인터넷 멀티캐스트 기술 발전에 가장 근접한 방식이라 할 수 있다.

PGM에서 사용되는 패킷 종류는 다음과 같다.

- ODATA(Original Content Data)
- NAK
- NCF(NAK Confirmation)
- RDATA(Retransmission Data)
- SPM(Source Path Message)

각 PGM 패킷은 해당 데이터의 송신자 및 세션 정보 파악을 위해 TSI(Transport Session Identifier)를 포함한다. 따라서 PGM 라우터 및 수신자들이 여러 개의 세션을 쉽게 구분할 수 있다. NAK 패킷 만 수신자 측에서 송신자 측으로 전달되며(upstream), 나머지 패킷은 모두 downstream으로 전달된다.

PGM은 실시간 서비스 제공을 위해 "TW
(Transmit Window)”을 사용한다. 이것은 sliding window의 방식으로, 만약 TW 동안 송신자가 NAK 메시지를 받지 못하면, 해당 데이터에 대한 복구가 사실상 불가능하게 된다.

PGM은 전체적으로 NAK 기반 오류제어를 수행한다. 따라서 송신자에게 돌아오는 NAK의 수를 줄이는 것이 유일한 이슈가 된다. 여기에는 라우터의 도움이 필요하다. 먼저 데이터 손실을 탐지한 수신자는 가까운 PGM 라우터에게 NAK를 전송하고, NAK 메시지는 라우터에서 라우터로 멀티캐스트 트리를 따라 이동한다. 이러한 NAK 전달은 NCF 혹은 RDATA 메시지가 발견될 때까지 계속된다. NCF와 RDATA는 이미 해당 메시지의 재전송 요청이 다른 수신자에 의해 이루어졌음을 의미한다. 이러한 경우 PGM 라우터들은 송신자에 이르는 upstream 경로를 따라 중복(duplicate) NAK 메시지를 제거한다.

NAK 메시지가 송신자에게 전달되는 경로는 송신자에서 수신자에게 이르는 멀티캐스트 트리 경로와 동일해야 한다. 이를 위해 ODATA 패킷 전송 시에 SPM 메시지가 삽입(interleave)되어 각 수신자에게 전달된다. PGM 라우터는 이 정보를 송신자로의 NAK 메시지 포워딩을 위한 사용한다. SPM은 또한 TW 안에 있는 가장 오래된 패킷이 곧 TW에서 없어질 것이라는 정보를 각 수신자에게 알려주는 역할도 한다. 따라서 송신자는 적어도 TW가 전달되는 속도만큼으로 SPM을 전송해야 한다(데이터 패킷 하나당 적어도 하나의 SPM 메시지가 전송되어야 한다).

PGM 라우터는 어디에서 NAK가 오는지를 알기 때문에, RDATA의 전송경로 혹은 전송포트를 실제 NAK 메시지 출처로 재현할 수 있다. 따라서 불필요한 재전송이 일어나지 않는다.

정리하면, PGM은 일종의 수신자주도방식의 오류제어 방식이다. PGM은 네트워크 계층의 멀티캐스트 트리를 이용하여 오류제어를 수행한다. 재전송 트리의 구성 혹은 서버의 설치 등의 별도 오버헤드 없이 오류제어를 수행할 수 있다. 한번 실시간응용에 부합하도록, TW 범위 안에서만 오류를 복구한다. TW 기간의 데이터는 영구적인 손실로 간주한다. 결과적으로 간단하면서도 매우 높은 확장성을 달성할 수 있다.

하지만 PGM은 오직 실시간 전송에 적합하고, 이에 따라 영구적인 데이터 손실이 가능한다는 장점이 단점으로 작용한다. 또한 PGM의 구현을 위해서는, 네트워크 계층의 모든 라우터들이 PGM 기제의 메커니즘을 처리할 수 있는(비록 요구되는 메커니즘이 매우 적다고 해도) 능력이 요구된다.

IV. 결론 및 향후 연구방향

지금까지 기존에 발표된 신뢰성 멀티캐스트 프로토콜들의 오류제어기법에 대하여 살펴보고, 그 특징에 따라 5가지로 분류하였다. 송신자 주도 오류복구기법은 분명히 확장성에 많은 문제를 지니고 있다. 확장성 문제를 해결하기 위해 수신자주도 혹은 토론 기반의 오류복구기법이 제안되었으나, 남아가 전체 수신자 그룹을 서버 혹은 트리 기반으로 그룹화하여 로컬 복구를 추구하는 기법들이 최근에 개발되고 있다. 한편, 응용서비스마다 다른 신뢰성 요구를 가진다는 점에 착안하여, 특정 응용서비스에 적합한 오류제어기법에 대한 개발 연구도 한창 진행중이다.

기존 오류제어기법에 대한 비교와 분석결과를 토대로 우리는 다음과 같은 결론을 얻을 수 있다.

*첫째, NAK 기반 수신자주도방식이 ACK 기반 수신자주도방식보다는 더 좋은 확장성을 갖는다.*
하지만, 호름체어를 위해 어떤 형태로든 ACK 정보가 송신자에 전달되어야 한다. 이를 위한 방식으로 주기적 폴링 방식이 좋은 대안으로 여겨진다.

또한 기반 오류재생방식은 토큰 사용자를 중심으로 트래픽 부하를 분산시키는 효과를 얻을 수 있지만, 근본적인 확장성 문제를 해결해주지는 않는다.

전체 수신자로 로컬그룹으로 조직화하여 오류 복구를 수행하는 서버 기반 혹은 트리기반방식은 수신자주도 혹은 토큰 기반 오류재생방식에 비하여 더 좋은 확장성을 제공한다.

모든 응용 서비스에 적합한 오류재생방식을 개발하기란 매우 어렵다. 따라서 특정 응용에 적합한 방식을 개발하는 것도 좋은 대안이다.

이러한 분석을 토대로, 확장성을 위한 오류재생 기법으로는 트리 기반 로컬복구방식이 적합할 것으로 여겨진다. 또한 기본적으로는 NAK를 사용하며, 주기적인 폴링을 통해 송신자에게 ACK 정보를 전달할 필요가 있다. 하지만 PGM과 RMTP 방식의 비교에서 알 수 있듯이, 로컬 복구를 위한 트리로써, 제어용 논리적 트리와 실제 네트워크 계층의 멀티캐스트 트리 중첩, 어느 것이 더 효율적인지 아직 판단할 수 없다.

한편, 대부분의 기존 연구에서는 QoS(Quality of Service) 및 그룹관리 측면에 대한 고려가 부족한 편이다. 특히 응용서비스가 요구하는 시간간 혹은 패킷 손실률 등의 정보가 오류 제어 메커니즘에 포함되어 있지 않다. 따라서 향후에 개발되는 신뢰성 멀티캐스트 프로토콜에서는, 이러한 QoS 및 그룹관리 정보를 충분히 반영한 오류제어 기법이 제안되어야 한다.


참 고 문헌


