신기술 해설

Survey on Reliable Multicast over the Internet

이 동 만*, 윤 원 용**

● 목 차 ●

1. 서 론
2. NACK 억제 프로토콜
3. 트리 기반 프로토콜
4. FEC 기반 프로토콜
5. 표준화 동향
6. 이슈
7. 결 론

1. 서 론

1990년대 말 이후, 급격한 발전과 확산을 거듭하고 있는 인터넷과 웹의 성공은 사용자들의 욕구를 충족시키는 다양한 컨텐트의 개발이 그 주된 요소이지만 그 성공의 이면에는 "신뢰적" 전송 프로토콜인 TCP라는, 십여 년간 연구되고 검증되어 온 프로토콜이 있었기에 가능한 것이었다. TCP [39]는 비신뢰적인 IP 계층 위에 신뢰성을 더함으로써, 현재의 대다수 인터넷 응용 프로그램 (WWW, FTP, TELNET 등)을 지원하는 트랜스포트 프로토콜로 광범위하게 사용되고 있다. 그러나 TCP는 일대일 유니캐스트 통신만을 지원하므로, 점차 그 중요성이 부각되고 있는, 다수의 사용자 또는 프로세스들이 참여하는 인터넷 응용들 (주문형 비디오, 원격 교육, 인터넷 방송, 주식 정보 등)을 지원하기에는, 네트워크 대역폭이나 호스트 처리율 면에서 비효율적인 한계점을 지닌다.

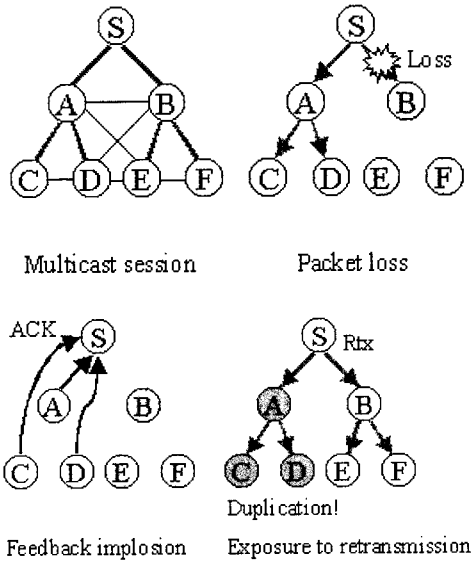
이러한 새로운 다중 사용자 인터넷 응용들을 지원하기 위해서는, 패킷 복제 기능을 지닌 라우터를 이용하여 효율적으로 데이터 패킷을 전달하는 IP 멀티캐스트 [8] 위에, 신뢰성 있는 전달 서비스를 제공하는 신뢰적 멀티캐스트 (reliable multicast) 프로토콜들을 사용하는 것이 효율적이다.

* ICU 부교수
** ICU 박사과정 수료

1.1 신뢰적 멀티캐스트 문제

신뢰성 (reliability)이란 데이터가 송신자에서 수신자로 손실 없이 전달됨을 일컫는다. 인터넷과 같이 비신뢰적인 네트워크 상에서 신뢰성을 제공하는 과정은 손실 감지 (loss detection)와 손실복구 (loss recovery) 두 단계로 이루어진다. 유니캐스트 전송의 경우, 패킷의 손실은 송신측에서 수신자의 ACK을 받지 못한 채 타이머가 종료될 때 또는 수신측에서 순서번호의 공백을 감지하고 NACK을 보내어 송신자가 손실 사실을 알게 되는 순간 (loss detection), 유니캐스트 재전송하면 된다 (loss recovery). 그러나 멀티캐스트 전송의 경우, 어느 노드가 패킷 손실을 감지하고 알릴 것인가, 어느 노드가 재전송을 어떻게 할 것인가의 문제가 복잡해지며, 이 문제를 단순히 취급하면, 예를 들면, TCP의 에러 제어 방법을 단순히 멀티캐스트로 확장하는 것 따위, 대단히 비효율적일 수 있는 위험이 있다.

그림 1의 네트워크에 노드 S를 송신자로 하고 노드 A에서 F까지의 수신자로 구성된 멀티캐스트 세션이 존재하고, S를 루트로 하는 해당 멀티캐스트 라우팅 트리가 파란색 굵은 실선으로 표시되어 있다.[1] TCP와 유사하게, 송신자가 수신자의 집합을 관리하고 패킷 손실을 감지하며 재전송을 담당하는 이른바 송신자 주도 (sender-initiated) 프로토콜 [46]의 문제점을 묘사한다. 송신자는 패킷을 송신하면 타이머를 동작시키는데 이 타이머가 종료되기 전에 모든 수신자로부터 유니캐스트 ACK을 받지 못하면 패킷을 멀티캐스트 재전송한다.[2] 멀티캐스트 패킷을 성공적으로 수신한 노드들은 모두 (A, C, D) 송신자에게 ACK을 보내야 하므로 송신자와 그 주변 네트워크에 피드백 정보가 밀집되는 이른바 피드백 폭주 (feedback implosion) 문제가 발생한다. 송신자와 그 주변 네트워크를 병목지점으로 만들어 세션 전체의 성능을 감소시킬 수 있다. 손실 감지 단계의 implosion 문제에 이어 손실 복구 단계에서는 재전송 노출 (exposure to retransmission) 문제 [35]가 뒤따른다. 송신자가 재전송을 할 때, 이미 수신한 노드들 (A, C, D)이 재전송 패킷을 중복적으로 수신하기 때문에, 불필요한 처리 오버헤드와 네트워크 대역폭 낭용을 초래한다. 송신자 주도 프로토콜은 R명의 수신자가 있을 때, 모든 수신자들이 성공적으로 하나의 패킷을 받기 위해서는 $O(\ln R)$번의 재전송이 필요하다 [46]. 따라서, 수신자의 exposure는 $O(\ln R)$의 복잡도를 지닌다. 각 수신자는 같은 복잡도 만큼 송신자에 ACK을 보내므로 송신자의 implosion은 $O(R\ln R)$의 복잡도를 지닌다 (표 2 참조).



(그림 1) 멀티캐스트 세션 및 송신자 주도 프로토콜.

Implosion과 exposure 문제는 멀티캐스트 세션 참여자 수가 증가할수록 그 해악은 더욱 심각해지기 때문에, 세션의 확장성을 위해서는 이 두 문제를 효과적으로 해결할 수 있어야 한다.

1.2 접근방법

1.2.1 ARQ (Automatic Repeat reQuest) 방식

신뢰적 멀티캐스트의 가장 기본적인 접근방법은 손실된 패킷을 재전송하는 ARQ 방식이다. 전술한 바와 같이 송신자 주도 ARQ 방식은 implosion/exposure 문제가 심각하므로, 이를 해결하기 위한 다양한 방법이 제안되었다. ACK implosion을 줄이

1) 네트워크 계층의 멀티캐스트 라우팅 트리는 DVMRP, MOSPF, PIM-DM 등과 같은 소스 트리 방식 프로토콜 또는 CBT, PIM-SM 등의 공유 트리 방식 프로토콜에 의해 생성된다. 멀티캐스트 라우팅 알고리즘 및 프로토콜에 관한 개관은 [44]을 참조하면 된다.

2) Levine [19]은 송신자 주도의 의미를 송신자가 손실 감지를 한다는 것에 두지 않고 ACK을 받고 데이터를 메모리에서 해제할 수 있는 것에 두고 있다. 타이아웃 메커니즘은 송신자 뿐 아니라 수신자에서도 수행될 수 있음을 지적한다. 즉 수신자는 일정 비율로 ACK을 보내고, 일정기간 동안 패킷을 못 받으면 NACK을 보내는 것이 가능하므로 송신자가 손실을 감지한다는 사실은 송신자 주도의 충분조건이 되지 못한다.

기 위해, 수신자 주도 (receiver-initiated) 방식은 수신자가 패킷 손실을 감지한다. 수신자는 패킷 순서 번호의 갭을 보고 손실을 알거나 일정 기간동안 패킷을 받지 못하면 손실 났음을 감지하고 NACK을 보내어 재전송을 요청한다. NACK은 유니캐스트될 수도 있고, 멀티캐스트되어 NACK억제 (suppression)를 꾀할 수도 있다. 송신자가 수신자들의 집합을 몰라도 되므로 ACK 처리 로드가 없는 점도 확장성 향상에 기여한다. 수신자 주도 방식의 대표적인 예는 SRM [9]과 NORM [1]을 들 수 있다. 트리 기반 방식은 수신자들로 이루어진 계층적인 ACK 트리를 이용하는데, 트리의 부모 노드는 자식 노드들의 신뢰적 데이터 전달을 책임지고, 자식 노드는 ACK을 송신자가 아닌 자신의 부모에게 보내므로, ACK implosion을 피할 수 있다. 한 노드가 맡아야 할 자식 노드의 수는 전체 수신자 수에 독립적인 상수로 유지 가능하므로, 확장성 측면에서 큰 장점을 지닌다 [19]. RMTP [24], RMTP-II [48], TMTP [51], TRAM [7], LORAX [18] 등이 대표적인 예이다. 링 구조를 이용하여 ACK을 전달하는 방식 (예: RMP [47])도 있으나, 확장성이 떨어지는 것으로 알려져 있다.

1.2.2 FEC (Forward Error Correction) 방식

FEC 방식은 중복된 데이터를 미리 제공하여 일부가 손실되더라도 수신측에서 자체 복구 가능하도록 하여 수신자로부터의 피드백 (ACK 또는 NACK)을 없앤다. 제한적인 경우 FEC 자체로 implosion 및 exposure 없이 데이터 복구가 가능하지만 [4], 대부분 멀티캐스트 응용의 경우 ARQ 방식과 상호 보완적으로 사용되어 확장성 있는 신뢰적 멀티캐스트를 지원할 수 있다. [33]는 두 가지 형태를 제시하는데 하나는 ARQ 밑 계층에 FEC가 위치하여 손실 확율을 경감시키는 역할을 하는 것과 FEC과 ARQ를 하나의 계층에 통합하여 복구를 효율적이게 하는 것이다.

1.2.3 라우터 지원 (Router-assisted) 방식

라우터 지원 방식은 ARQ 또는 FEC 방식의 성능을 향상시키기 위해 네트워크 라우터에 특별한 기능을 설치하는데, 가령, exposure를 최소화하기 위해 멀티캐스트 라우팅 트리의 서브 트리로 멀티캐스팅이 가능하도록 하는 것을 들 수 있다 [24][35]. 능동망 (active network)의 노드에 ACK 결합과 NACK 필터링 기능을 구현하는 방법도 제안되었다 [5]. 라우터 지원 방식은 기존 인터넷상의 모든 멀티캐스트 라우터에 새로운 기능 추가가 되어야 제대로 동작하는 전개 (deployment) 문제가 있다.

<표 1> 신뢰적 멀티캐스트 프로토콜의 접근방법

	End-to-end			Router-assisted	
Tree-based ARQ	RMTP-II, TRACK	TMTP, TRAM, LORAX		RMTP	LMS, QTERS
FEC		Digital Fountain, ALC			
NACK-based ARQ	SRM		NORM	PGM	

1.3 글의 구성

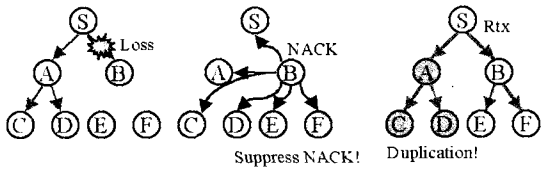
1990년대 이후, 신뢰적 멀티캐스트 문제에 대해 수 많은 방법들이 제안되어 왔지만 [34], 가장 유력한 인지도를 지닌 것은 NACK 억제 방식, 트리 기반 방식, FEC 기반 방식으로 압축되고 있다. 이 세 가지 방식에 초점을 맞추어, 2절과 3절에서 두 가지 ARQ 방식 (NACK 억제 프로토콜과 트리 기반 프로토콜), 4절에서 FEC 기반 프로토콜의 기본 동작 원리와 대표적인 예를 알아본다. 5절에서 IETF 표준화 동향을 살펴본다. 6절에서는 아직 해결되지 못한 신뢰적 멀티캐스트 이슈들을 정리하고, 마지막으로 7절에서 결론을 맺는다.

2. NACK 억제 프로토콜

전술했듯이, 기본적인 수신자 주도 프로토콜은

손실을 감지한 수신자들이 각각 NACK을 유니캐스트하는데, 이 경우 NACK implosion 문제가 발생한다. 이를 방지하기 위해, NACK 억제 (suppression) 기법을[3] 채택한 수신자 주도 프로토콜이 제안되었다. NACK을 멀티캐스트하여, 다른 수신자가 NACK을 보내는 것을 제한하도록 하여 NACK 개수를 최소화하는데, 이상적인 경우 패킷 손실 하나에 대해 하나의 NACK이 전송되는 것을 목표로 한다.

손실을 감지한 수신자들은 손실 감지 후 타이머를 가동시키고 종료할 때까지 일단 기다리는데, 그동안 손실을 먼저 감지한 다른 참여자가 보낸 NACK을 받는 경우, 보내려고 준비했던 NACK을 취소하고 자신은 NACK을 보낸 것으로 간주하여 NACK 타이머를 가동시키고 그렇지 않으면 실제로 자신이 NACK을 보내고 NACK 타이머를 가동시킨다. NACK 타이머의 종료 후에도 재전송 패킷이 도착하지 못하면 다시 NACK 전송을 시도한다. 그림 2의 노드 B, E, F가 손실을 감지하고 NACK 전송을 기다리는데, 노드 B의 타이머가 먼저 종료되어 NACK을 멀티캐스트하고, 이 NACK을 받은 노드 E, F는 준비하던 NACK을 취소한다. 이런 식으로 NACK implosion 문제를 상당히 줄일 수 있으나, 재전송 패킷이 멀티캐스트되기 때문에 노드 A, C, D는 중복적으로 패킷을 수신하게 되는 exposure 문제가 여전히 남아있다. NACK 억제 방식에서 R명의 수신자들이 모두 패킷을 성공적으로 수신할 때까지의 재전송 회수는 송신자 주도 방식과 마찬가지로 $O(\ln R)$이고, 따라서 수신자의 exposure의 복잡도는 $O(\ln R)$이다. 하지만, 피드백은 이상적인 경우 하나의 NACK만 송신자에 이르므로, 총 $O(\ln R)$의 송신자 implosion이 발생한다 (표 2 참조). NACK 억제 방법으로 송신자 주도형에 비해 implosion의 복잡도가 낮아졌음을 알 수 있다 [46].



(그림 2) NACK 억제 프로토콜

다른 수신자 주도 방식과 마찬가지로, 송신자가 수신자의 수신확인을 받지 않기 때문에 언제 버퍼를 해제해야 될 지 모르는 문제가 있다. 교착상태를 피하기 위해서는 무한대의 버퍼가 필요하다.[4]

2.1 SRM (SCALABLE RELIABLE MULTICAST)

SRM은 공유 화이트보드 (shared whiteboard)와 같은 다자간 응용을 지원하기 위해 개발된 것으로 실제 MBone에서 실험되고 있다 [9]. SRM은 수신자 주도 방식 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 억제 기법을 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 타이머의 경우 2_i $[C_1 d, (C_1+C_2)d]$ 구간에서 임의의 (random) 값을 선택하도록 한다. d는 수신자 자신과

3) 이러한 부류를 RINA (Receiver-Initiated with NACK-Avoidance)라 부른다. 제어 트래픽 양을 제한하는 방법은 XTP [6]의 'slotting and damping' 알고리즘이 처음이었고 후에 TMTP [51], SRM [9]이 비슷한 방식을 도입한다.

4) 현재의 수신자 주도형 프로토콜은 명시적인 버퍼 해제 메커니즘이 없다. 참고로 SRM의 멀티캐스트되는 주기적인 세션 메시지는 소스로부터의 가장 최근에 수신한 패킷의 순서번호를 포함하므로 버퍼 해제에 사용될 수 있음에 유의한다 [19].

송신자와의 추정된 RTT 값으로, 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 $[D_1 d, (D_1+D_2)d]$ 구간의 랜덤한 값을 선택한다. 여기서 d는 재전송하는 참여자와 재전송 요청을 한 참여자간의 추정된 거리이다.

초기 LAN을 위한 NACK 억제 기법과 달리 SRM은 세션 메시지 교환에 의해 추정된 지연 시간 값에 기반하여 타이머를 적응시켜 WAN에서의 성능을 향상시키고자 하였다. 그러나 타이머를 설정하기 위해 지연시간을 추정하는 것은 세심한 주의를 요하며 주기적인 세션 메시지의 교환 또한 만만치 않은 작업이다. 특히 세션 멤버쉽이 동적인 상황과 가변적인 네트워크 상황에서 최적의 타이머 값 설정이 어렵다.

SRM은 NACK 억제 기법으로 implosion 문제를 어느 정도 해결할 수 있지만, 이는 복구 시간의 지연이라는 대가를 치러야 한다. 또한, 재전송 패킷이 세션 전체로 멀티캐스트되므로 소수의 참여자만이 손실을 겪은 상황에서는 exposure 문제가 심각하다. 이에 대해 Liu 등은 NACK과 재전송 패킷을 멀티캐스트할 때, TTL 값을 설정하여 손실을 겪은 수신자만이 재전송 패킷을 받을 수 있도록 하는 지역 복구 방식을 제시하였다 [25]. 하지만 TTL 메커니즘 자체가 방향성이 없고 또한 TTL이 제한하는 범위는 정밀성이 떨어지기 때문에, exposure 문제를 완전히 배제하지는 못한다.

2.2 PGM

PGM (Pragmatic General Multicast) [45]은 Cisco Systems 사와 WhiteBarn 사가 개발하고 있는데, 멀티캐스트 신뢰성 및 서브캐스팅 기능을 지원하는 PGM-enabled 라우터를 이용한다. 수신자 주도 방식과 마찬가지로, 손실을 겪은 수신자가 NACK을 피드백하는데, 중간에 PGM-enabled 라우터들이 NACK을 집성함으로써 송신자의 NACK implosion을 막을 수 있다. 수신자가 NACK을 가까운 PGM-enabled 라우터, 또는 PGM NE (Network Element)에게 유니캐스트하면, PGM NE는 NCF (NACK Confirmation)을 수신자에게 보내고, NACK은 송신자를 향해 업스트림 PGM NE로 전달한다. NACK은 PGM NE들을 거쳐 송신자에게 도달한다. PGM NE는 서브트리에서 이미 NACK을 받았으면, 새로 도착한 NACK을 전달하지 않는다. SRM에서는 수신자들이 타이머를 이용하여 NACK을 억제하지만, PGM에서는 멀티캐스트 라우터의 기능을 이용하여 NACK을 억제한다고 볼 수 있다. 송신자가 재전송하면, PGM NE는 이전에 NACK을 들었던 인터페이스로만 선택적으로 재전송 패킷을 전달한다. 송신자 대신 DLR (Designated Local Repair)가 재전송할 수 있는 옵션이 있고, FEC 옵션 사용도 가능하다. NACK의 경로가 데이터 패킷이 멀티캐스트되는 트리와 같도록, SPM 메시지가 주기적으로 전송되면, SPM 메시지를 받은 PGM NE는 이를 기억해 NACK 포워딩 때 참조한다.

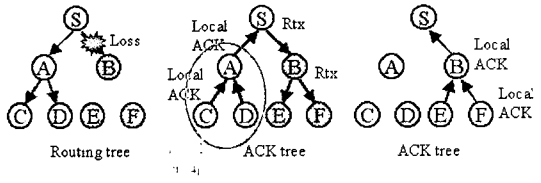
최근 Talarian 사가[5] PGM 프로토콜을 바탕으로 SmartPGM이라는 멀티캐스트 제품을 구현하는 등, PGM은 산업 표준으로 자리잡아 가고 있는 듯 하다. PGM-enabled 라우터의 전개 (deployment) 문제가 제약점이다.

3. 트리 기반 프로토콜

수신자들은 송신자가 루트인 ACK 트리 속에 계

5) http://www.talarian.com

층적으로 포함되어, 부모 노드와 자식 노드들의 지역 그룹 (local group)들이 가상적으로 형성된다. 이 지역 그룹 내에서 부모 노드는 송신자 주도 방식으로 자식 노드들로부터 ACK을 받고, 재전송을 담당한다. 지역 그룹 내에서의 피드백과 재전송 방법은 프로토콜마다 다르게 채택되어 있다. 그림 3에서, 노드 A, C, D로 이루어진 지역 그룹, 노드 B, E, F로 이루어진 지역 그룹, 노드 S, A, B로 이루어진 지역 그룹이 있다. 송신자는 전체 수신자가 아닌 자신의 자식 노드의 ACK만 수신하므로, ACK implosion을 피할 수 있다. 재전송 시, 송신자 S는 자식 B에게 재전송을, 노드 B는 다시 자신의 자식 E와 F에게 재전송을 수행한다. 따라서, 이미 성공적으로 수신한 노드 A, C, D는 재전송 패킷을 중복적으로 받지 않는다. 즉, 트리 방식은 implosion뿐만 아니라, exposure 문제도 최소화할 수 있는 장점을 지닌다. R명의 수신자가 있을 때, 송신자가 받는 ACK의 개수는 R과는 무관하게 자식 수에 좌우되므로, 자식 수가 B (상수)라고 할 때, implosion의 복잡도는 $O(B)=O(1)$이다 [19]. 수신자의 exposure은, 부모 노드가 자신의 모든 자식 노드들이 패킷을 수신할 때까지의 재전송 회수가 $O(\ln B)$ 이므로, 역시 $O(1)$의 복잡도를 지닌다. 표 2에 ARQ 방식의 implosion 및 exposure의 복잡도를 정리해 놓았는데, 트리 프로토콜이 가장 높은 확장성을 지닌다.



(그림 3) 트리 기반 프로토콜.

<표 2> ARQ 방식의 복잡도 분석

	implosion	exposure
Sender-initiated protocols	$O(R\ln R)$	$O(\ln R)$
NACK suppression protocols	$O(\ln R)$	$O(\ln R)$
Tree-based protocols	$O(1)$	$O(1)$

3.1 RMTP (RELIABLE MULTICAST TRANSPORT PROTOCOL) 및 RMTP-II

Paul [36]은 최초로 트리 기반 프로토콜을 제시하였는데 이것은 뒤에 RMTP로 개발되었다. RMTP [24]는 정적인 ACK 트리 구조를 이용하여 신뢰적인 멀티캐스트를 서비스한다. 수신자들을 각자의 국소 지역 (local region)에 속하게 하고 지역마다 지정된 DR (Designated Receiver)로 하여금 지역 복구를 담당하도록 하여 확장성 향상을 꾀한다. 수신자는 송신자에 ACK을 보내는 것이 아니라 주기적으로 ACK을[6] 지역의 DR로 유니캐스트함으로써 implosion 문제를 피할 수 있다. 그러나 DR이 정적으로 결정되어 있어야 하는 취약점을 내포하고 있으며, 주기적인 ACK은 손실 감지 즉시 재전송 요청을 하지 못하기 때문에 지연시간이 증가된다.

재전송은 손실을 겪은 지역내 수신자 수에 따라 유니캐스트 또는 멀티캐스트되는데 멀티캐스트 시 대역폭을 효율적으로 사용하기 위해 손실을 겪은 수신자 수가 많은지 확인을 위한 기다림이 필요하므로 지연 문제가 있다. 지연을 없애기 위해 즉시 유니캐스트할 수 있으나 이 경우 대역폭 증가라는 반대급부가 있다.

수신자가 자신의 DR을 결정하는 과정은 다음과 같다. 송신자와 모든 DR이 동일한 TTL 값을 지니는 제어 패킷을 주기적으로 보내면 수신자는 가장 높은 TTL 값의 - 따라서 가장 가까운 - DR을 자신의 DR로 선택한다. DR은 DR 선정 제어 패킷이나 재전송 패킷을 멀티캐스트할 때 지역(자신의 서브트리)로만 보낼 수 있도록 하는 서브 트리 멀티캐스팅 (sub-tree multicasting) 기법을 이용한다. DR이 세션의 송신자를 거짓으로 소스로 설정한 멀티캐

6) 현재까지의 수신한 최대 순서번호와 손실정보를 나타내는 비트맵을 담고 있어서 보통의 패킷당 ACK과는 다르며 ACK과 NACK의 역할을 동시에 하는 것으로 보아야 한다.

스트 패킷을 유니캐스트 패킷에 캡슐화하여 라우터에 보내면, 라우터는 멀티캐스트 패킷을 분리하여 마치 송신자에게서 온 것인 양 다시 멀티캐스트한다. 이것은 멀티캐스트 라우터의 수정을 요하는데 RMTP의 제약점으로 볼 수 있다.

RMTP-II [48]는 RMTP를 기반으로 하는데, 보다 높은 확장성과 낮은 지연시간을 위해 NACK 기반 복구와 FEC 기반 복구 방식을 옵션으로 채택하고 있다. 그러나, 여전히 RMTP와 마찬가지로, ACK 트리가 정적으로 정해져 있어야 하는 단점이 남아 있다. Talarian 사는 1999년 인수한 GlobalCast 사의 신뢰적 멀티캐스트 기술과 자사의 미들웨어 기술에 기반하여 1999년 11월, 최초의 상업 신뢰적 멀티캐스트 제품인 RMTP-II를 발표하였고 SmartSockets(기술을 접목하고 있다. 현재 Sprint 등의 통신회사와 결합하여 미국 주요 네트워크에 DR 서버를 구축하고, 기업체 고객을 대상으로 멀티캐스트 서비스의 제공을 계획하고 있다.

3.2 TMTP (TREE-BASED MULTICAST TRANSPORT PROTOCOL)

TMTP [51]는 ERS (Expanded Ring Search) 기법을 사용하여 물리적 트리에 가까운 논리적 트리를 구성한다. 논리적 제어 트리상의 노드는 K개의 자식 노드의 신뢰성만 책임을 지며 자식 노드는 부모에게만 주기적으로 ACK을 유니캐스트한다. ACK을 보냄으로써 송신자 또는 부모 노드의 버퍼를 해제할 수 있게 하되 주기적 전송의 이유는 implosion 현상을 줄이기 위함이다. 송신자 또는 부모 노드는 그 자식으로부터 ACK을 받지 못한 채 타임아웃되면 제한된 범위의 멀티캐스트 (limited scope multicast)를 통해 재전송한다. 이러한 송신자 주도의 속성 외에 TMTP는 빠른 복구를 위해 NACK에 기반한 복구를 병행한다. 자식 노드가 손실을 감지하면 NACK을 억제 기법을 쓰는 제한된 범위의 멀티캐스트로 보내고 부모는 TTL을 세팅하여 제한된 범위의 멀티캐스트를 이용하여 지역 복구를 수행한다. 그러나 NACK 중복을 막기 위한 지연 시간이 단점이다. SRM에서와 마찬가지로 TTL 메커니즘에 기반한 지역 복구는 exposure 문제를 완전히 배제하지 못한다.

논리적 트리는 ERS 메커니즘에 의해 구성된다. 새로운 노드가 트리에 가입할 때 제어 패킷을 TTL 값을 점차 증가해 가며 멀티캐스트하는데, 가장 작은 값에서 응답한 노드의 자식으로 들어간다. TTL의 무방향성 (lack of direction)과 멤버의 가입순서에의 종속성으로 인해 ACK 트리를 하부 멀티캐스트 라우팅 트리와 유사하게 유지하는 것이 매우 어렵다.

3.3 기타 트리 기반 프로토콜

LBRM (Log-Based Receiver-reliable Multicast)은 특별히 분산가상환경 (Distributed Virtual Environment)를 위해 고안된 프로토콜로서 로깅 서버 (logging server)의 계층적 트리 구조로 신뢰성 책임을 분산시킨다 [14]. 각 서버들은 데이터를 무한대로 저장하고 해당 서버를 통해 수신자가 손실을 복구하도록 하는 방식이다. 송신자는 로깅 서버로부터 ACK을 받으면 메모리에서 데이터를 축출할 수 있다. 수신자가 NACK을 서버에 보내면 서버는 NACK의 수를 보고 재전송을 유니캐스트로 할 지 멀티캐스트로 할 지 결정한다. 부 (secondary) 로깅 서버는 지역 복구를 담당하고 자신이 복구할 수 없는 경우에는 주 (primary) 로깅 서버에게 필요한 데이터를 요청한다.

LGMP (Local Group based Multicast Protocol)는 ACK 처리와 손실 복구의 부담을 분산시키기 위해 지역 그룹 (local group)의 개념을 도입하였다 [12][13]. 지역 그룹은 고유한 멀티캐스트 주소를 가지며 각 지역 그룹에는 그룹 제어자 (group controller)가 존재한다. 지역 그룹의 수신자는 그룹 제어자에게 주기적으로 자신의 수신 상태 정보를 알리며 그룹 제어자는 각 수신자들의 상태 정보를

규합하여 주기적으로 그룹의 상태 정보를 송신자에게 보낸다. 그룹 제어자는 지역 그룹내의 재전송을 담당한다. 그룹 제어자 자신이 손실을 겪은 경우에는, ACK을 보낸 멤버에게 유니캐스트하거나 그룹 전체로 멀티캐스트하여 손실된 패킷을 요청하며 그 패킷을 이미 수신한 멤버는 중복 억제 정책을 써서 그룹으로 멀티캐스트하여 그룹 제어자의 손실을 복구한다. 그룹 제어자의 손실에 대비하여 그룹 제어자의 ACK이 올 때까지 수신자들은 자신이 수신한 데이터를 버퍼에서 축출하면 안 된다. 그룹 제어자들이 트리 구조로 구성될 수 있지만, 구체적인 트리 생성/유지 방법은 제시되어 있지 않고, 이용할 수 있는 정보만을 제공한다.

TRAM [7]은 Java™ Reliable Multicast Service™ 프로젝트의 일환으로 Sun Microsystems 사에서 개발되고 있다. 서버 즉 DR의 존재를 가정하는 RMTP와는 달리, 서버 없이 오직 수신자들만으로 ACK 트리를 구성하도록 하고 있다. 양방향으로 멀티캐스트가 가능한 경우 뿐 아니라 단방향으로만 멀티캐스트가 가능한 경우에도 ACK 트리를 구성할 수 있게 하며, LAN 기반 트리 구성, ACK 트리 최적화 등 풍부한 트리 구성/관리 방법을 제시하고 있다.

3.4 ACK 트리 구성 방법

트리 기반 프로토콜의 제약점은 ACK 트리를 멀티캐스트 라우팅 트리와 유사하게 구성하게 될 때, 최대의 성능을 보인다는 점인데, 세션의 멤버쉽이 동적이고 (수신자가 자주 가입/탈퇴) 하부 라우팅 경로가 동적으로 변할 수 있는 상황에서 그러한 ACK 트리를 구성하고 유지하는 것은 대단히 어려운 일이다 [20][50]. RMTP나 LBRM은 정적인 트리를 채택하기 때문에 동적인 환경에 대한 적응성이 없다.

Tracer [20]는 IGMP MTRACE 기능을 이용하여 송신자까지의 경로를 추적함으로써, 라우팅 트리상의 가까운 상위 노드를 결정적으로 (deterministically) 알아 낼 수 있다. 경로의 통고(path advertisement)와 응답을 교환하여 적절한 부모를 선택할 수 있다. MTRACE 기능을 네트워크의 일부에서 관리 정책적으로 제공하지 않을 때, 동작하지 않을 수 있다. ERS를 이용한 주기적인 경로 통보는 그 오버헤드가 크다. 소스 노드는 각 수신자가 보내는 IGMP MTRACE 질의에 유니캐스트로 응답해야 하므로 소스가 병목이 될 수 있다.

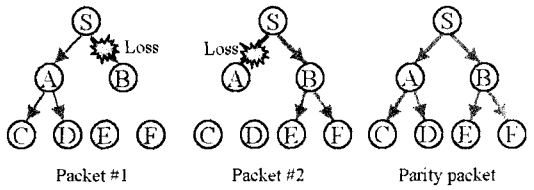
LMS [35]는 DMCAST (Directed Multicast)라는 새로운 라우터 전달 서비스를 이용하여, 응답자 혹은 재전송 노드의 계층 구조를 만든다. 새로운 IP 옵션, 라우팅 엔트리 및 IGMP 리포트에 대한 변경 등 상당한 양의 변경 요구사항이 뒤따른다. Tracer나 LMS와 같은 네트워크 계층의 방식은 서브 트리 멀티캐스트를 가정하는 RMTP와 마찬가지로 네트워크 전개에 어려움이 있다.

라우터의 도움이 필요 없는 종단간 방법 (end-to-end approach)도 있다. S. Ratnasamy [40] 등은 수신자들이 관찰한 loss print 정보를 이용하여, "true shared losses"의 내림 차순으로 수신자들을 묶어 가면서 하나의 트리로 구성해 나간다. 새로운 노드가 기존 트리에 참여해야 될 때, 적당한 자리를 찾기 위해서는 기존 모든 수신자들과 "true shared loss" 확률을 계산해야 하므로 그 복잡도가 상당하다.

4. FEC 기반 프로토콜

FEC 기반 프로토콜은 중복된 데이터, 즉 패러티 패킷 (parity packet)을 미리 송신하여 수신측에서 데이터 패킷의 손실을 복구할 수 있도록 함으로써, 피드백 implosion 문제를 해결한다. 피드백 없이 신뢰 전송이 가능하므로, 위성 회선과 같은 비대칭적 (asymmetric) 망에서 효과적이며 지연시간이 적다는 장점이 있다. 에러 감지/교정을 위한 특별한 코딩이 필요한데, ATM 망에서 주로 많이 사용되는

Reed-Solomon 코드 [23]와 소프트웨어로 구현될 때 수십 배 빠른 Tornado 코드 [26]등이 있다. 인코딩/디코딩 지연 시간과 구현의 복잡함을 단점으로 들 수 있다.



(그림 4) FEC 프로토콜

FEC방식이 특별히 멀티캐스트에 도움이 되는 점은 패러티 패킷 하나가 서로 다른 패킷의 손실을 겪은 수신자들의 손실 패킷 복구에 이용될 수 있다는 것이다 [32]. ARQ가 서로 다른 패킷들의 모두 즉 수퍼셋 (superset)을 재전송해야 하는 점과 구분된다. 그림 4에서 노드 A, C, D는 첫번째 패킷을, 노드 B, E, F는 두 번째 패킷을 손실하였으나, 패러티 패킷 하나만을 수신하여 각각 손실되었던 패킷을 복원할 수 있다. ARQ 방식이라면, 두 번의 패킷 재전송이 필요하다.

대표적인 예로 Digital Fountain [4], SHARQFEC [16], proactive FEC [43] 등을 들 수 있다.

4.1 DIGITAL FOUNTAIN

FEC 기반 프로토콜은 n개의 데이터 패킷에 k개의 패러티 패킷을 함께 보내어, 총 n+k 개 패킷 중 어느 것이라도 n 개 이상의 패킷을 수신하는 경우 n 개의 데이터 패킷을 복원할 수 있도록 한다. 기존의 대부분의 FEC 방식은 Reed-Solomon 코드를 이용하였는데, 이 코드는 n이 굉장히 클 때 인코딩/디코딩 시간이 오래 걸려 k의 크기를 낮춰야 하는 단점이 있어서, 송신자는 보내야 할 데이터가 클 경우에는 이를 작은 블록들로 나누어 블록마다 FEC을 적용한다 [33]. 블록 내에서 충분한 패킷 수를 받지 못하면 복원이 불가능하므로, 수신자는 재전송을 요청하여 한다. 즉 재전송을 완전히 배제할 수는 없었다. 다수의 수신자들이 서로 다른 시각에 데이터를 수신할 때는, 동일한 블록이 여러 번 전송되어야 하는 단점이 있다.

Digital Fountain은 송신할 데이터 파일 전체 (가령 n 개 패킷)를 Tornado 코드 [26]를 써서 무한 개의 패킷으로 인코딩함으로써, 수신자는 어느 시점이든 원하는 시점에 들어와서 n개의 패킷을 받고서[7] 원본 데이터 파일을 복원할 수 있도록 한다. 재전송이 필요 없으며, 송신자는 비동기적 (asynchro-nous) 수신자들에 대해 따로 같은 패킷을 중복 전송할 필요가 없다. Reed-Solomon 코드가 체 (field) 연산과 행렬 역변환 (matrix inversion) 등을 사용하는 반면, Tornado 코드는 XOR 연산만 사용한다. 또, Reed-Solomon 코드는 패러티 패킷 하나를 생성할 때 n 개의 데이터 패킷 중 많은 수를 이용하지만, Tornado 코드는 상대적으로 적은 수를 필요로 하므로 디코딩 시간이 훨씬 줄어든다.

Digital Fountain 사는[8] Meta-ContentTM (Digital Fountain using Tornado codes) 기술에 기반한 Download Fountain™이라는 제품을 출시하였다. 하나의 서버로 TV 규모인 수백만 명의 사용자들이 음악, 비디오, 전자 출판 등의 인기 있는 대용량 파일을 동시에 다운로드할 수 있다. 동일한 기술에 기반한 Streaming Fountain™은 고품질 비디오를 on-demand로 청취할 수 있는 제품이다.

5. 표준화 동향

지난 10여년간의 집중적인 연구로 신뢰적 멀티

7) 이상적인 digital fountain 의 개념에 따르면, 정확히 n개의 패킷을 수신하면 되지만, Tornado 코드를 써서 구현하는 경우, 실제 n개보다 약간 많은 수의 패킷을 수신하여야 디코딩이 가능하다.

8) http://www.digitalfountain.com

캐스트 분야는 어느덧 표준화를 거론할 정도의 충분히 성숙한 단계에 이르렀다. 1999년 3월 IETF RMT 워킹 그룹이[9] 결성되어, 확장성 있는 일대다 멀티캐스트 프로토콜과 구조의 표준화 작업을 진행해오고 있다 [11][49]. 하나의 멀티캐스트 프로토콜이 다양한 종류의 멀티캐스트 응용들의 요구사항을 모두 충족시키기는 어렵기 때문에 각 응용마다 특정한 성격의 멀티캐스트 프로토콜들이 필요할 것으로 보고, 먼저 여러 프로토콜들에 공통적으로 사용될 수 있는 요소를 BB (Building Block)으로 정의하고, 실제 프로토콜은 이들 BB의 조합으로 정의하는 방법 (PI: Protocol Instantiation)으로 표준화를 추진 중이다. 현재 GRA (Generic Router Assist) BB, NACK BB, FEC BB, 혼잡 제어 BB, 트리 구성 BB, 보안 BB 등이 거론 중이다. PI를 보면, 앞서 정리한 세 가지 유력한 방식 (NACK 억제, 트리 기반, FEC 기반 방식)이 각각 NORM, TRACK, ACL로 표준화 진행 중이다.

5.1 NORM (NACK-ORIENTED RELIABLE MULTICAST)

NORM PI [1]은 기본적으로 NACK 억제 기법을 채택하고 이외에 메시지 집성 (aggregation) 기능과 패킷 레벨 FEC을 더하여, 피드백 implosion을 줄이고, 사전 (proactive) 전송 옵션을 제공한다. 송신자는 혼잡 제어, RTT 추정, 수신자 동기화 등을 위해 제어 메시지를 생성한다. 또한 송신자가 ACK을 요구할 때는 수신자들이 ACK을 피드백할 수 있는 옵션을 가지고 있다.[10] 이 정보는 송신자의 버퍼 관리나 혼잡 제어에 사용될 수 있다. 자세한 프로토콜 동작은 아직 미정 상태이다 [1].

9) http://www.ietf.org/html.charters/rmt-charter.html

10) 이 경우, 엄격히 따지자면 NORM은 NAPP (Negative Acknowledgment Periodic Polling) 부류에 속하게 되어, [19]의 정의에 따르면 송신자 주도형으로 분류될 수 있다.

5.2 TRACK (TREE-BASED ACK)

TRACK PI [50]은 트리 기반 프로토콜로서, Talarian 사의 RMTP-II와 Sun Microsystems 사의 TRAM을 기반으로 표준화가 진행되고 있다. ACK 트리는 트리 생성 BB [15]에 의해 생성된다. 기본적으로, ACK 트리의 자식 노드는 ACK을 RH (Repair Head)로 불리기도 하는 부모 노드에게 보내고, 부모는 멀티캐스트를 통해 재전송하도록 한다. ACK 트리의 지역 그룹 내에서 NACK 기반 복구와 FEC 기반 복구를 수용하기 위해 NACK BB [2]와 FEC BB [29]을 사용할 수도 있다.

5.3 ALC (ASYNCHRONOUS LAYERED CODING)

FEC 기반 ALC PI 는 멀티미디어 멀티캐스트에서 계층 코딩을 이용한 혼잡 제어 방식을 빌려온 것으로서, 송신자는 FEC을 써서 데이터 패킷과 패리티 패킷을 인코딩하여 여러 개의 멀티캐스트 계층 (또는 그룹)으로 보내면 수신자는 자신의 처리 능력에 따라 다른 계층에 가입할 수 있다 [27][28]. 즉, 에러 제어와 함께 수신자 기반 다중 전송률 (multi-rate) 혼잡 제어가 동시에 이루어진다.

6. 이슈

6.1 다대다 신뢰적 멀티캐스트 (MANY-TO-MANY RELIABLE MULTICAST)

인터넷 멀티캐스트 응용은 비디오/오디오 데이터의 일대다 비신뢰적 전송에서, 파일 전송, 경제정보 등의 일대다 신뢰적인 전송으로 그 흐름이 변화하고 있다. 앞으로는 협동 작업, 네트워크 게임 등의 다대다 신뢰적인 전송이 새로운 멀티캐스트 응용으로 등장할 것으로 보인다.

과거 신뢰적 멀티캐스트 프로토콜에 대한 대부분의 연구 개발은 일대다 전송에 초점을 두었고, 일대다 전송에서의 해결책이 다대다 전송에도 효

과적으로 사용되리라고 가정하였다 [37]. 그러나 다대다 전송은 전혀 다른 특성을 지닌다. 간단한 예를 든다. 하나의 패킷을 신뢰적으로 전송하는데 송신자가 처리해야 할 일 또는 시간 (implosion이 대부분을 차지)을 X라 하고, 수신자 처리 시간 (exposure가 대부분을 차지)을 Y라 하자. X=100, Y=10인 프로토콜 A가 있다. 일대다 세션의 병목 지점의 처리 시간은 X와 Y중 큰 값 100이 되고, 세션의 경우, 세션의 처리율 (throughput)은 이 값의 역수이다. A와는 다르게 X를 50으로 줄이는 대신 Y가 20으로 늘어난 프로토콜 B를 생각하자. 프로토콜 B의 일대다 세션 처리 시간은 50으로 줄어든다. 이제 다대다 경우를 생각하자. N명으로 구성된 다대다 세션에서 한 노드는 하나의 패킷을 송신하고, 나머지 N-1 명으로부터 각각 하나씩 패킷을 수신하므로, 세션의 처리시간은 X+(N-1)Y가 된다. 프로토콜 A의 다대다 세션 처리시간은 N=11일 때, 200이 되는 반면, 프로토콜 B의 경우 250으로 오히려 증가한다. 일대다 세션에서 우수한 프로토콜이 다대다 세션에서는 나쁜 성능을 보일 수 있음을 알 수 있다. [52]은 일대다 세션의 경우는 implosion을 줄이는 것이, 다대다 세션의 경우는 exposure를 줄이는 것이 멀티캐스트 프로토콜의 확장성을 높이는 방안임을 보여 준다. 현재까지 제시된 프로토콜 중 다대다 응용을 목표로 한 것은 SRM [9], RMP [47], LORAX [18]등을 들 수 있으나, 애초에 위의 지적을 고려해서 설계된 것은 아니다. 기존의 프로토콜 중 다대다 세션에서 가장 확장성 있는 것은 무엇인지, 새로운 접근방법이 존재할 것인지에 대한 이슈가 남아 있다.

다대다 응용이 반드시 확장성을 요구하지도 또 반드시 다대다 신뢰적 멀티캐스트 세션을 요구하는 것도 아니다. 화상회의나 협동작업 응용의 경우 사람들의 수는 실생활에서처럼 상호작용이 가능한 인원으로 제한될 것이다. 수천 명 이상으로 이루어진 분산가상환경 (distributed virtual environment)에서 참여자는 모든 데이터에 관심이 있는 것은 아니다. 인접함과 관심도 (interest)에 따라 신뢰적으로 전송되어야 하는 데이터가 일부에 국한된다. 이런 경우, 다대다 멀티캐스트 프로토콜을 쓰는 것보다 하나의 응용을 여러 개의 작은 세션으로 분할하는 것이 효과적인데, 이 분할 방법이 이슈가 된다 [17][21].

6.2 혼잡 제어 (CONGESTION CONTROL)

혼잡 제어 문제는 에러 제어 문제에 비해 그 시작이 다소 늦어졌으나, 에러 복구 기술이 어느 정도 안정적인 단계에 다다르자, 실제 인터넷에 성공적으로 사용되기 위해 꼭 필요한 요소로 주목을 받고 있는 실정이다. 네트워크에 혼잡 상황이 발생할 때 이에 맞춰 전송률을 낮춤으로써 혼잡 상황을 빨리 벗어날 수 있도록 하는데, 기존의 인터넷 트래픽의 대부분을 차지하는 TCP 트래픽과 공정하게 네트워크 대역폭을 사용할 수 있어야 한다. 혼잡 상황을 빨리 감지하고 대응할 수 있어야 하면서, 동시에 확장성에 저해가 되지 않는 피드백 기법을 필요로 한다. 주로 IRTF RMRG 미팅에서 발표와 의견 교환이 이루어져 왔는데, TCP-friendly한 혼잡 제어 방식이 지지를 받았으며, AIMD (Additive Increase Multiplicative Decrease) 보다 공격적이어서는 안되며 네트워크 변화에 빨리 적응할 수 있어야 한다는 요구사항에 합의하였다. 멀티캐스트 트리 경로의 다양함으로 인해 수신자의 손실정보에 일일이 대응할 경우 송신율이 0 가까이로 떨어지는 이른바 "loss path multiplicity"또는 "drop-to-zero" 문제와 가장 느린 수신자에 의해 세션의 전송률이 결정되는 문제는 대표적인 어려운 문제로 각각 UMASS [3]와 이탈리아 Pisa 대학에서 해결책을 제시한 바 있다 [42]. Bell 랩에서 전송률 기반 혼잡 제어와 윈도우 기반 혼잡 제어를 비교하는 연구 [10]와 NCSU 대학에서 트리 구조를 이용한 윈도우 혼잡 제어 기법도 제시되었다 [41]. 이러한 연구에

도 불구하고 혼잡 제어 문제는 아직 에러 복구처럼 이론적인 증명이나 합의가 이루어지지 않은 단계이다.

6.3 보안 (SECURITY)

1998년 결성된 IRTF의 Secure Multicast research group (SmuG)과 2000년 12월 시작된 IETF의 MSEC 워킹 그룹에서 연구와 표준화가 진행 중이다.[11] 그룹 키를 이용한 데이터 암호화 및 송신자 인증, 그룹 보안연합 관리 및 암호화 키 관리 등의 연구를 진행하고 있다. 현재 이슈로는 소스만을 신뢰할 수 있는 상황에서 어떻게 데이터의 무결성 (integrity)와 인증 (authenticity)을 입증할 수 있을 것인가, 그룹 멤버쉽이 동적으로 변할 때 어떻게 공통의 그룹 키를 유지할 것인가의 문제를 들 수 있다 [30].

6.4 이동 망에서의 신뢰적 멀티캐스트 (MOBILE RELIABLE MULTICAST)

이동 컴퓨팅의 발전은 기존의 유선 망과 무선 이동 망이 결합된 차세대 인터넷을 예견하고 있다. 멀티캐스트 세션이 고정 호스트와 이동 호스트로 함께 이루어진 경우 새로운 문제점들이 부각될 수 있다. 무선 네트워크는 상대적으로 낮은 대역폭과 높은 오류 발생 율을 지니므로 피드백과 재전송을 효과적으로 제어할 필요가 있다. 이동 단말기는 상대적으로 낮은 성능과 제한된 배터리 용량을 가지므로 이동 단말기에 프로토콜 처리에 따른 부하를 많이 가해서는 안 된다. 또한 단말기의 핸드오프는 멀티캐스트 라우팅 경로의 변화와 함께 패킷의 중복 전달 또는 패킷의 손실을 자주 야기할 수 있다. 최근 유선 망의 신뢰적 멀티캐스트는 기존의 방식을 쓰고 이동 망의 기지국간에 이동 단말기의 신뢰성을 책임지는 논리적 링 (Logical ring) 프로토콜이 제시된 바 있으나 [31], 이동 IP 멀티캐스트 표준 [38]의 특성을 고려하지는 못했다.

7. 결 론

분명한 기술적인 장점에도 불구하고 과거 멀티캐스트 응용들이 널리 개발되지 못한 것은 ISP (Internet Service Provider)들이 멀티캐스트 트래픽의 폭주를 제어할 수 있는 방법을 갖지 못한 기술적 문제도 있지만 멀티캐스트 응용 시장의 조성이 이루어지지 않았던 점도 중요한 이유이다. 그러나, 최근 인터넷 사용자의 욕구는 단순한 일대일 통신에서 벗어나 상호작용이 가능한 다자간 통신으로 관심이 옮겨가고 있으며, 대역폭을 많이 사용하는 (bandwidth-intensive) 응용들은 기술적, 경제적 이유로 멀티캐스트 통신에 기반하지 않을 수 없다.

현재 예측되는 대부분의 멀티캐스트 응용들은 - 원격교육 (사이버 대학), 원격 공동작업 등과 같은 일반적인 그룹 통신 서비스로부터 뉴스, 광고, 메일 등의 응용 서비스에 이르기까지 - 신뢰적인 데이터 전달을 요구할 것으로 생각된다. 신뢰적 멀티캐스트 기술은 가까운 미래에는 실시간 증권 및 재무 정보를 신뢰적으로 전송하는 서비스, 대용량 파일 전송, 소프트웨어 갱신 등의 일대다 분배형 (one-to-many dissemination) 응용에서, 분산가상환경 (distributed virtual environment), 네트워크 게임, 실시간 고대역폭 분산 데이터베이스 갱신, 프록시/미러링/캐싱 사이트 등의 다대다 상호작용형 응용에까지 다양하게 사용될 것으로 전망한다. 단, 하나의 신뢰적 멀티캐스트 프로토콜이 이러한 다양한 응용의 요구조건을 모두 충족시킬 수는 없으므로, 유니캐스트 경우의 TCP와 같은 범용 프로토콜을 기대하기는 어려우며, 특정 응용을 위해 조율된 다수의 전용 프로토콜들이 사용될 것으로 보인다.

11) http://www.securemulticast.org

참고문헌

[1] B. Adamson, C. Bormann, M. Handley, and J. Macker, "NACK-Oriented Reliable Multicast Protocol (NORM)," Internet draft, draft-ietf-rmt-pi-norm-02.txt, July 2001

[2] B. Adamson, C. Bormann, M. Handley, and J. Macker, "NACK-Oriented Reliable Multicast (NORM) Protocol Building Blocks," Internet draft, draft-ietf-rmt-norm-bb-02.txt, July 2001

[3] S. Bhattacharyya, D. Towsley, and J. Kurose, "The Loss Path Multiplicity Problem in Multicast Congestion Control," IEEE INFOCOM '99, 1999.

[4] J. W. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A Digital Fountain Approach to Reliable Distribution of Bulk Data," ACM SIGCOMM '98, September 1998

[5] M. Calderon, M. Sedano, A. Azcorra, and C. Alonso, "Active Network Support for Multicast Applications," IEEE Network, May/June 1998

[6] G. Chesson, "The Evolution of XTP," Proc. IFIP Int. Conf. High Speed Networks, pp. 15-24, March 1991

[7] D.-M. Chiu, S. Hurst, M. Kadansky, and J. Wesley, "`TRAM: A Tree-based Reliable Multicast Protocol," Technical Report, SML TR-98-66, Sun Microsystems, July 1998

[8] S. Deering, "Host Extensions for IP Multicasting," IETF RFC 1112, August 1989

[9] S. Floyd, V. Jacobson, C. Liu, S. McCanne, and L. Zhang, "A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing," ACM SIGCOMM 95, pp. 342-356, August 1995

[10] S. J. Golestani and K. Sabnani, "Fundamental Observations on Multicast Congestion Control in the Internet," IEEE INFOCOM '99, 1999

[11] M. Handley, S. Floyd, B. Whetten, R. Kermode, L. Vicisano, and M. Luby, "The Reliable Multicast Design Space for Bulk Data Transfer," IETF RFC 2887, August 2000

[12] M. Hoffman, "A Generic Concept for Large-scale Multicast," Int. Zurich Seminar on digital communications (IZS '96), pp. 95-106, February 1996

[13] M. Hoffman, "Enabling Group Communication in Global Networks," Global Networking '97, pp. 321-330, November 1996

[14] H. W. Holbrook, S. Singhal, and D. R. Cheriton, "Log-based Receiver-Reliable Multicast for Distributed Interactive Simulation," ACM SIGCOMM '95, pp. 328-341, August 1995

[15] M. Kadansky, D. M. Chiu, B. Whetten, B. N. Levine, G. Taskale, B. Cain, D. Thaler, and S. Koh, "Reliable Multicast Transport Building Block: Tree Auto-Configuration," IETF Internet Draft, draft-ietf-rmt-bb-tree-config-02.txt, March 2001

[16] R. G. Kermode, "Scoped Hybrid Automatic Repeat reQuest with Forward Error Correction (SHARQFEC)," ACM SIGCOMM '98, pp. 278-289, August 1998

[17] E. Lety and T. Turletti, "Issues in Designing a Communication Architecture for Large-scale Virtual Environments," NGC '99, pp. 54-71, November 1999

[18] B. N. Levine, D. B. Lavo and J. J. Garcia-Luna-Aceves, "The Case for Reliable Concurrent Multicasting Using Shared Ack Trees," ACM Multimedia '96, pp. 365-376, November 1996

[19] B. N. Levine and J. J. Garcia-Luna-Aceves, "A

Comparison of Reliable Multicast Protocols," ACM Multimedia Systems, 6, pp. 334-348, 1998

[20] B. N. Levine, S. Paul, and J. J. Garcia-Luna-Aceves, "Organizing Multicast Receivers Deterministically by Parcket-Loss Correlation," ACM Multimedia '98, pp. 201-210, September 1998

[21] B. N. Levine, J. Crowcroft, C. Diot, J. J. Garcia-Luna-Aceves, and J. Kurose, "Consid-eration of Receiver Interest for IP Multicast Delivery," IEEE INFOCOM '00, pp. 470-479, March 2000

[22] D. Li and D. R. Cheriton, "OTERS (On-Tree Efficient Recovery using Subcasting): A Reliable Multicast Protocol," IEEE ICNP '98, pp. 237-245, October 1998

[23] S. Lin nad D. J. Costello, Error Control Coding: Fundamentals and Applications, Prentice-Hall, Englewood Cliffs, NJ, 1983

[24] J. C. Lin and S. Paul, "RMTP: A Reliable Multicast Transport Protocol," IEEE INFOCOM 96, March 1996

[25] C.-G. Liu, D. Estrin, S. Shenker, and L. Zhang, "Local Error Recovery in SRM: Comparison of Two Approaches," IEEE/ACM Transactions on Networking, 6(6), pp. 686-699, December 1998

[26] M. Luby, M. Mitzenmcher, A. Shokrollahi, D. Spielman, and V. Stemann, "Practical Loss-Resilient Codes," ACM Symposium on theory of Computing, 1997

[27] M.Luby, J.Gemmell, L.Vicisano, L.Rizzo, M.Handley, and J. Crowcroft, "Layered Coding Transport: A massively scalable multicast protocol," Internet draft, draft-ietf-rmt-bb-lct-00.txt, November 2000

[28] M.Luby, J.Gemmell, L.Vicisano, L.Rizzo, and J. Crowcroft, "Asynchronous Layered Coding: A massively scalable reliably content delivery protocol," Internet draft, draft-ietf-rmt-pi-alc-02.txt, July 2001

[29] M.Luby, L.Vicisano, J.Gemmell, L.Rizzo, M.Handley, and J. Crowcroft, "RMT BB Forward Error Correction Codes," Internet draft, draft-ietf-rmt-bb-fec-03.txt, July 2001

[30] M. J. Moyer, J. R. Rao, and P. Rohatgi, "A Survey of Security Issues in Multicast Communications," IEEE Network, pp. 12-23, November/December 1999

[31] I. Nikolaidis and J. J. Harms, "Logical Ring Reliable Multicast Protocol for Mobile Nodes," IEEE ICNP '99, pp. 106-113, 1999

[32] J. Nonnenmacher and E. Biersack, "Reliable Multicast: Where to use FEC," Proc. PfHSN '96, October 1996

[33] J. Nonnenmacher, E. Biersack, D. Towsley. "Parity-Based Loss Recovery for Reliable Multicast Transmission," ACM SIGCOMM '97, September, 1997

[34] K. Obraczka, "Multicast Transport Protocols: A Survey and Taxonomy," IEEE Communications Magazine, pp. 94-102, January 1998

[35] C. Papadopoulos, G. Parulkar, and G. Varghese, "An Error Control Scheme for Large-Scale Multicast Applications," IEEE INFOCOM '98, pp. 1188-1196, March 1998

[36] S. Paul, K. Sabnani, and D. Kristol, "Multicast Transport Protocols for High speed Networks," IEEE ICNP '94, October 1994

[37] S. Paul, K. Sabnani, J. C. Lin, and S. Bhattacharyya, "Reliable Multicast Transport Protocol," Journal of Selected Areas in Communications, 15(3), pp. 407-421, April 1997

[38] C. Perkins, et. al., "IP Mobility Support," RFC2002, October 1996

[39] J. B. Postel, "Transmission Control Protocol," RFC 793, September 1981

[40] S. Ratnasamy and S. McCanne, "Scaling End-to-end Multicast Transports with a Topologically-sensitive Group Formation Protocol," IEEE ICNP '99, pp. 79-88, October 1999

[41] I. Rhee, N. Ballaguru, and G. N. Rouskas, "TCP: Scalable TCP-like Congestion Control for Reliable Multicast," IEEE INFOCOM '99, 1999

[42] L. Rizzo, "pgmcc: a TCP-friendly single-rate multicast congestion control scheme," ACM SIGCOMM 00, pp. 17-28, August 2000

[43] D. Rubenstein, J. Kurose, and D. Towsley, "Real-Time Reliable Multicast Using Proactive Forward Error Correction," NOSSDAV '99, 1999

[44] L. H. Sahasrabuddhe and B. Mukherjee, "Multicast Routing Algorithms and Protocols: A Tutorial," IEEE Network, 14(1), pp. 90-102, January 2000

[45] T. Speakman, D. Farinacci, S. Lin, A. Tweedly, N. Bhaskar, R. Edmonstone, R. Sumanasekera, and L. Vicisano, "PGM Reliable Transport Protocol Specification," Internet Draft, draft-speakman-pgm-spec-06.txt, February 2001

[46] D. Towsley, J. Kurose, and S. Pingali, "A Comparison of Sender-Initiated and Receiver-Initiated Reliable Multicast Protocols," IEEE Journal of Selected Areas in Communications, 15(3), pp. 398-406, April 1997.

[47] B. Whetten, S. Kaplan, T. Montgomery, "A High-performance, Totally Ordered Multicast Protocol," Workshop on Theory and Practice in Distributed Systems, pp. 33-57, September 1994

[48] B. Whetten and G. Taskale, "An Overview of Reliable Multicast Transport Protocol II," IEEE Network, 14(1), pp. 37-47, January/February 2000

[49] B. Whetten, L. Vicisano, R. Kermode, M. Handley, S. Floyd, and M. Luby, "Reliable Multicast Transport Building Blocks for One-to-Many Bulk-Data Transfer," IETF RFC 3048, January 2001

[50] B. Whetten, D.-M. Chiu, M. Kadansky, and G. Taskale, "Reliable Multicast Transport Building Block for TRACK," Internet draft, draft-ietf-rmt-bb-track-01.txt, March 2001

[51] R. Yavatkar, J. Griffioen, and M. Sudan, "A Reliable Dissemination Protocol for Interactive Collaborative Applications," ACM Multimedia '95, pp. 333-344, November 1995

[52] W. Yoon, D. Lee and H. Y. Youn, "On the Scalability of Many-to-many Reliable Multicast Sessions," Technical Report, CDSN-2001-TR011, ICU, April 2001, available at http://cds.icu.ac.kr/research/reports.asp

저자약력

이 동 만

1982년 서울대학교 컴퓨터공학 학사
1984년 KAIST 전산학 석사
1987년 KAIST 전산학 박사
1987년-1988년 KAIST post doc
1988년-1997년 Hewlett-Packard 책임연구원
1997년-현재 ICU 부교수
관심분야 : 네트워크 및 분산 시스템, scalable network architecture for distributed virtual environment, reliable multicast protocol, fault-tolerant group communication, layered multimedia multicast, web caching, collaborative computing framework
E-mail : dlee@icu.ac.kr

윤 원 용

1996년 서울대학교 컴퓨터공학 학사
1998년 서울대학교 컴퓨터공학 석사
1998년-1999년 ETRI 위촉연구원
1998년-현재 ICU 박사과정 수료
관심분야 : reliable multicast, multicast congestion control, multimedia streaming
E-mail : wyyoon@icu.ac.kr