

신뢰적 그룹통신을 위한 ECTP 설계 및 구현

박주영* · 정옥조* · 강신각*

*한국전자통신연구원 표준연구센터

Design and Implementation of ECTP for Reliable Group Communications

Juyoung Park* · Ok Jo Jung* · Shin Gak Kang*

*ETRI Protocol Engineering Center

E-mail : jypark@etri.re.kr

요 약

신뢰적으로 멀티캐스트 데이터를 전달하는 메커니즘을 설계하기 위해선 1:1 유니캐스트 환경과는 달리 고려되어야 할 주요한 사항들이 있다. ECTP-1은 IP멀티캐스트 데이터 전송과정에 신뢰성을 제공해 주는 트랜스포트 계층의 프로토콜로서, 계층적 제어 트리를 구성하여 신뢰적 데이터를 전송한다. ECTP-1은 현재 리눅스 환경에서 구현되었으며, mftp(multicast ftp)와 whiteboard 응용 프로그램을 사용하여, APAN 망 환경에서 성능을 검증하였다.

ABSTRACT

Reliable multicast data transmission in a 1:N environment needs more sophisticated error control mechanism than that of in 1:1 environment due to ACK implosion and duplicated retransmission. Although there have been many related research on error control in reliable multicast, real implemented protocols are rare. As one of the reliable multicast transport protocols, ECTP is selected as an international standard reliable multicast protocol by ITU-T and ISO and implemented on RedHat 7.2 machine by us. In this paper, we evaluate the performance of the error control mechanism in the respect of throughput and generated control packet numbers with a real implementation code. From the results, it is concluded that the suitable values of error control parameters can be obtained from the local group size and network environments.

키워드

Reliable Multicast Transport Protocol

1. 서 론

인터넷 Reliable Multicast는 다수의 수신자들에게 신뢰적인 데이터 전송을 가능케 하는 메커니즘이다. 그런데 1:1 환경과는 달리 1:N환경에서는 고려되어야 할 많은사항들이 있는데, 우선 하나의 세션에 얼마나 많은 수신자들이 참가할 수 있는지에 관한 확장성 제공 방안, 흐름제어, 오류제어 등을 손꼽을 수 있다[1]. 본 고에서는 이들 고려 사항들 중에서 오류 제어메커니즘에 대한 내용을 다루었다. 인터넷에서의 오류는 전송 경로에서의 패킷 변형에 의한 손실 보다는 손실에 의한 오류가 대부분을 차지하며, 이러한 오류들은 현재의 인터넷에서는 대부분 오류에 대한 재전송 요청과 재전송 방법

을 사용하여 복구한다.

그런데 1:N 멀티캐스트 환경에서 오류를 복구하기 위해서는 제어 메시지에 의한 폭주(ACK implosion)방지와 재전송 방법에 대한 세밀한 고려가 있어야 한다. ACK implosion은 다수의 수신자들이 하나의 송신자에게 제어 메시지를 동시에 보낼때 발생하며, 이때 송신자는 N개의 수신자들로 부터 급작스런 제어 메시지를 받게 된다. 이러한 ACK implosion 을 방지하기 위한 메커니즘이 각종 연구단체등에서 연구되어 왔으며, 가장 대표적인 방법들 중 하나는 제어메시지의 전달을 위한 계층적 제어 트리를 구성하여 오류 및 손실을 복구하는 방법을 들 수 있다. 또 다른 방법으로 계층적

제어 트리를 구성하는 대신 손실이 발생한 노드로부터 가장 근거리에서 있는 노드로부터 복구 데이터를 수신하는 방법을 들 수 있다. 한편 손실된 데이터를 재전송하기 위한 메커니즘도 각기 다르다. 계층적 트리를 구성하는 경우 상위 부모 노드로부터 재전송을 하는 프로토콜로부터 이웃에게서 재전송 데이터를 받거나, 아예 손실이 발생하였을 경우 복구할 수 있도록 잉여 복구 데이터를 동시에 보내는 방법도 제안 되었다.

ECTP-1[3]은 계층적 제어 트리를 구성하여 신뢰적 데이터를 전송하는 메커니즘을 사용하는데, 신뢰적인 데이터 전송에 필요한 제어 메시지의 수를 줄이거나 보다 효율적인 데이터 전송을 위해서 사용자가 조절할 수 있는 파라미터 집합을 제공한다.

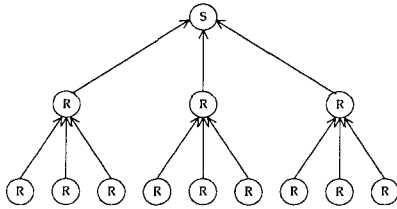


그림 4 ECTP 제어 트리 구성

ECTP-1의 시험을 위해서 mftf(multicast ftp) 응용 프로그램을 동작시켰다. 테스트 환경으로는 APAN 망을 이용하였다. APAN(Asia Pacific Advanced Network)은 1997년 6월에 여러 가지 선진적 응용 서비스 및 기술들을 국제적으로 협력 시험하고, 국가간의 협력사업을 목표로 아래 아태지역의 학계, 연구기관을 중심으로 구성된 컨소시엄으로써 현재 Korea, Australia, China, Hong Kong, Japan, Malaysia, Singapore, Taiwan 등의 8개국 주요 멤버로 활동하고 있으며, 미국, 스리랑카 등의 연합멤버들도 아울러 활동 중이다. 본 고에서는 ECTP-1의 기능 검증에 위해 한국과 일본의 APAN 멤버간 구동 시험 내용과 그 결과를 기술한다.

II. 신뢰성을 보장하기 위한 메커니즘

신뢰적인 멀티캐스트 전송 프로토콜에서 오류를 복구하기 위해서 크게 송신자가 오류를 복구할 수 있도록 데이터를 전송할 때 잉여 제어 데이터를 in-band 혹은 out-band로 전송해줌으로써, 수신자가 재전송 요청을 하지 않아도 스스로 오류 복구를 할 수 있도록 해주는 메커니즘(FEC: Forward Error Correction)이나 오류가 발생하였을 경우 수신자가 송신자에게 재전송을 요청하는 일반적인 오류 복구 방식인 ARQ(Automatic Repeat Request) 방식을 많이 사용한다.

FEC방식을 사용할 경우 송신자는 수신자들을 관리할 필요나 손실을 복구하기 위한 재전송 버퍼

를 관리할 필요가 없기 때문에 비교적 수신자들의 수가 큰 지역이나 Open-ended 멀티캐스트 그룹에서 사용되어진다. 그런데 이 방식은 수신자들이 오류를 복구할 수 있도록 잉여 데이터를 전송해야 한다는 것과 복구를 위한 복잡한 알고리즘이 필요하다는 단점도 있다.

반면에 ARQ방식은 송신자가 데이터를 제대로 수신했는지에 관한 수신자 상태 정보를 관리해야 하며, 아울러 재전송을 요청하였을 때, 데이터를 재전송해 주기 위한 재전송 버퍼를 관리해야 한다는 단점이 있지만, 자동 오류 복구를 위한 잉여 데이터를 보낼 필요가 없기 때문에, 단지 순수 데이터만을 전송하면 된다. 또한 불필요한 재전송을 막기 위한 정보관리를 통하여 수신자에 대한 관리가 용이하다.

ECTP-1프로토콜은 신뢰적인 전송을 제공해주는 멀티캐스트 전송 프로토콜으로써 오류를 복구하기 위해서 ARQ방식을 사용한다. 그런데 ARQ방식을 사용할 경우 다수의 수신자들로부터 동시에 재전송 요구등의 제어 메시지가 올 수 있기 때문에 ACK implosion의 문제가 발생할 수 있다.

이러한 ACK implosion을 방지하기 위해서 ECTP-1은 다음의 3가지 방법을 이용하여 동일 데이터 패킷에 대한 ACK를 최소화할 수 있는 방법을 제공한다.

■AGN (Ack Generation Number)

ECTP-1의 모든 수신자들은 송신자로부터 고유한 CID(Child ID)를 할당받는데, ECTP-1 송신자는 매핑된 CID와 수신자의 주소(IP주소와 Port) 정보를 관리함으로써, 수신자들은 그 세션에서 고유한 CID를 할당해줄 수 있다.

ECTP-1 세션은 임의의 AGN값을 갖는다. AGN값은 수신자들에게 언제 ACK를 보낼 수 있는지를 결정할 수 있게 하는데, 이 값을 통하여 ACK implosion을 줄일 수 있다. AGN을 통한 ACK 발생 규칙은 다음과 같다.

만일 패킷의 시퀀스 번호를 세션의 AGN값으로 나눈 나머지 and 시퀀스 번호를 수신 노드 고유의 아이디인 CID로 나눌때, 동일한 값을 나타낼 경우 ACK를 보내도록 한다.

만일 수신자가 3이고, AGN값이 8일 경우, 전송되는 데이터 수에 대해 발생하는 ACK 패킷의 수는 3/8이 되며, 수신자가 3이고 AGN값이 1일 경우 동일 데이터 패킷 수에 대하여 발생하는 ACK패킷의 수는 3배가 될 것이다.

■RXT(Retransmission Time)

ARQ에서 만일 어떤 데이터 패킷에 대한 일정한 시간 이내에 ACK가 도달하지 않을 경우 이를 ACK의 손실 혹은 전송한 데이터의 손실로 판단할 수 있어야 한다. ECTP-1에서는 송신자가 RXT 시간 동안 수신자들로부터 ACK를 받지 못했을 경우 이를 손실로 판정하고 해당 패킷들을 재전송한다.

■RBT (Retransmission Backup Time)

ECTP-1에서는 AGN을 이용하여 ACK의 수를 제한한다. 그런데 다수의 수신자들로 구성된 세션에서 데이터의 전달되는 중에 ACK를 발생함으로써 수신된 데이터를 제대로 반영하지 못하는 경우가 발생할 수 있다.

RBT 시간은 전송되는 도중에 ACK가 세션 전체에 올바르게 수신한 패킷에 대한 정보를 반영하지 못하는 경우를 완충하기 위한 파라미터로써, 비록 재전송 요청이 오더라도 해당 패킷을 전송한지 RBT시간 이내일 경우 재전송하지 않는다.

이 매커니즘을 통하여 송신되고 있는 데이터에 대한 불필요한 중복 재전송을 막을 수 있다.

III. 구현

ECTP-1 프로토콜 개체는 리눅스 환경에서 C언어로 구현되었다. 프로토콜의 동작은 [그림2에 보이는 바와 같이 사용자-레벨 형태로 동작된다.

ECTP는 응용 프로그램과 UDP 사이에서 신뢰적인 멀티캐스트 데이터 전송을 보장하는데, 응용과 UDP 사이에는 IPC(Inter Process Communication) 방식을 이용하였다. ECTP응용들을 위해서 ECTP는 버클리 소켓과 유사한 자체적인 API를 msocket 이라는 이름으로 제공한다. msocket 라이브러리는 응용과 함께 컴파일되어 ECTP-1의 서비스를 이용할 수 있게 해주는데, 사용방법은 버클리 소켓 라이브러리의 그것과 유사하다.

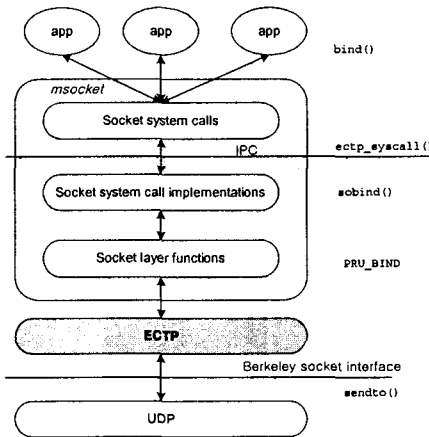


그림 5 ECTP 구현 구조

ECTP-1 프로토콜 스택은 응용의 요청을 해석할 수 있는 so_msocket 라이브러리와 프로토콜 엔진으로 크게 구분할 수 있는데, so_msocket 부분은 socket system call 부분과 socket layer function으로 구성된다.

응용 계층의 msocket 라이브러리는 응용 프로그램의 요청을 메시지열로 변환하여 이를 ECTP 프

로토콜 스택으로 `ectp_syscall()` 함수를 사용하여 메시지열 형태로 전달한다.

IV. 시험

APAN 환경에서 ECTP-1의 동작을 검증하고자, 다음 [그림3]에서와 같은 한-일간 시험망을 구축하였다. 국내에서는 충남대학교, ETRI가 참가하였으며, 일본에서는 규슈대학교가 참가하였다.

멀티캐스트를 지원하기 위해 각 기관간의 연결은 다음과 같다. 충남대-규슈대, 충남대-KOREN 간은 DVMRP 터널링을 사용하였으며, ETRI-KOREN 간은 PIM을 사용하였다. 그러나 ETRI의 라우터에 설정된 PIM과 충남대와 규슈대의 라우터에 설정된 DVMRP가 KOREN에서 접속될 때 불안정한 상태를 나타내어 부득이 실험 대상을 충남대와 규슈대에서만 수행하였다.

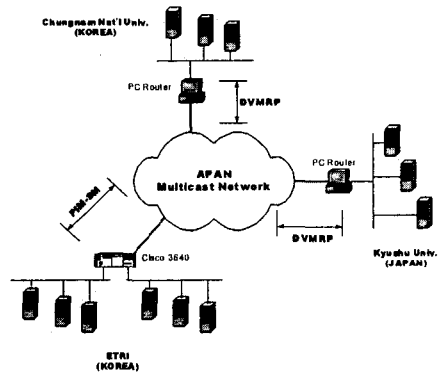


그림 6 APAN을 이용한 시험 환경 구축

실험의 목적은 ECTP-1을 로컬 테스트베드가 아닌 순수한 멀티캐스트 망환경에서 사용할 경우에 신뢰적인 데이터의 전달과 프로토콜의 오버헤드를 검사하는 것이다.

신뢰적인 데이터의 전달을 검증하기 위해선 일정한 크기의 압축 파일을 ECTP-1 그룹으로 전달한 뒤 수신자 노드에서 수신한 파일과 송신 파일을 비교함으로써 신뢰적인 데이터 전달이 되는 것을 확인 할 수 있다.

프로토콜의 오버헤드는 일정한 량의 데이터를 보낼때 얼마만큼의 추가적인 정보가 필요한가를 나타내는 척도로써, 헤더의 크기, 프로세서 자원 할당 및 사용되는 제어 메시지의 량을 꼽을 수 있겠다.

그런데 멀티캐스트 환경에서는 일반 1:1 환경과는 달리 다수의 수신자들로 이루어진 그룹 환경에서의 통신이기 때문에 신뢰적인 데이터를 전달하기 위해 필요로 하는 제어 메시지의 량이 1:1 환경의 그것에 비하여 무척 클 수 밖에 없다.

따라서 본 실험에서는 데이터 트래픽과 제어 메시지 트래픽을 측정함으로써 ECTP-1의 프로토콜

오버헤드를 가늠하고자 한다.

실험을 위한 토폴로지로서 3개의 경우를 고려하였다. 환경1) 일본측의 하나의 송신자와 한국에서의 2개의 수신자로 이루어진 경우([그림4]), 환경2) 하나의 일본 측 송신자와 세 개의 한국측 수신자로 이루어진 환경([그림5]), 환경3) 한국측에서 하나의 송신자와 두개의 수신자, 일본측에 하나의 수신자로 이루어진 경우([그림6])를 고려하였다.

[그림4]는 환경1에서 ECTP-1을 이용해 파일을 전송할 때의 트래픽량을 나타낸 그림이다. 그림에서 사각형으로 표시된 선은 신뢰적 데이터 전송을 위해 필요로 하는 제어 메시지의 량을 나타낸 것이며, 마름모꼴로 표시된 것은 데이터의 양을 표시한 것이다. x축은 파일 전송에 소요된 시간, 그리고 y축은 각 메시지들의 양을 바이트 단위로 표시하였다.

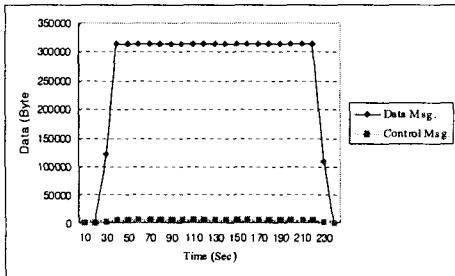


그림 7 환경1에서의 트래픽 측정

[그림4]에서 데이터 전송이 이루어진 시간은 20초에서부터 230초까지이다. 이때 데이터 트래픽에 대한 제어 메시지 트래픽들의 비율은 약 1~2% 정도를 보인다는 것을 알 수 있다. [그림5]에서는 약간의 제어 메시지들이 증가한 것을 볼 수 있는데, 이 이유는 수신자들이 증가함으로써 ACK 메시지가 증가하는데 기인한 것이다.

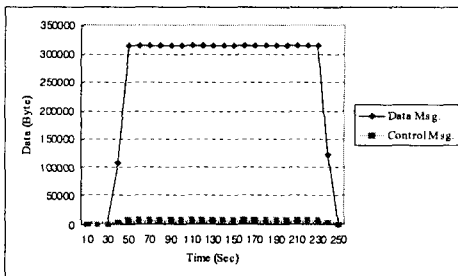


그림 8 환경2에서의 트래픽 측정

[그림6]은 비록 송신자의 위치가 달라진다고 하더라도, [그림4]와 [그림5]에 비해 그 프로토콜 오버헤드가 크게 달라지지 않았음을 보여준다.

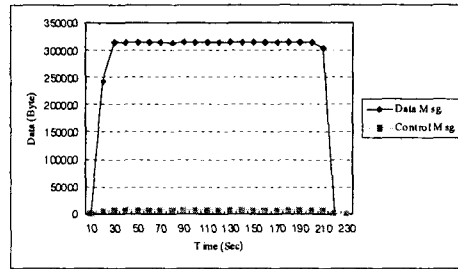


그림 9 환경3에서의 트래픽 측정

V. 결론

지금까지 ECTP-1의 개요, 구현 및 한-일간 APAN 멀티캐스트 환경을 이용한 ECTP-1 시험내용을 소개하였다. ECTP-1은 IP 멀티캐스트 환경에서 그룹에서 1:N 신뢰적인 데이터를 전달해 주는 프로토콜인데, 시험 결과를 통해 ECTP-1이 로컬 시험망이 아닌 순수한 IP 멀티캐스트 망에서도 무리없이 동작한다는 사실을 증명하였으며, 현재의 설계가 그룹 파일 전송 등에 별 수정 없이 사용될 수 있으리라 사료된다. 향후 시험망의 범위를 확장시키도록 하여 좀더 광범위한 실험으로 확대할 예정이다.

참고 문헌

- [1] IETF RMT WG, "http://www.ietf.org/charter/rmt-wg.html"
- [2] ISO605|ITU 13252, ECTS(Enhanced Communications Transport Services)
- [3] ITU-T 14476-1, ECTP-1 Part1
- [4] ECTP Official Homepage, "http://ectp.etri.re.kr"
- [5] ECTP-1, ITU-T Recommendation X.606 | ISO/IEC 14476-1