

ECTP-3: Implementation and Experimentations

2006년 12월

경북대학교 통신프로토콜연구실

박재성 (knucsid@gmail.com)

Abstract

본 문서는 ECTP-3 (ITU-T X.607 | ISO/IEC 14476-3) 표준규격에 의거한 프로토콜 구현코드 및 테스트 응용프로그램 실험에 관한 기술문서이다.

<첨부파일: ectp3.tar.bz2>

Table of Contents

1. 서론	2
2. ECTP 개요	2
2.1 ECTP-3 RELATED TERMS	2
2.2 ECTP-3 프로토콜 개요	3
3. ECTP-3 APIS	4
4. 구현코드 및 설치	6
4.1 LIBRARY FOR TEST APPLICATIONS.....	6
4.1.1 Mplayer.....	6
4.1.2 Qt Library.....	7
4.2 ECTP-3 SOURCE COMPILE	7
4.2.1 File Structures	7
4.2.2 Compile.....	8
5. 실험 결과	9
5.1 TEST SCENARIO.....	9
5.2 RESULTS.....	10
6. 결론	10
REFERENCE	11

1. 서론

인터넷 시대가 열리면서 최근 IPTV, 인터넷 교육 방송 등의 양방향성을 지향하는 응용들이 많이 생겨나게 되었다. 기존의 유니캐스트를 기반으로 하는 응용들은 사용자가 많으면 많을 수록 네트워크 자원을 많이 소비하게 된다. 그래서 이것을 해결하기 위한 프로토콜이 필요하게 되었다. ECTP(Enhanced Communications Transport Protocol)는 인터넷 멀티캐스트 응용을 지원하기 위한 프로토콜이다. 특히 ECTP-3는 양방향성을 지원하는 멀티캐스트 연결을 지원하는 프로토콜이다. 이 프로토콜은 다량의 사용자가 사용하더라도 일정한 네트워크 자원을 소비함으로써 유니캐스트 기반보다 훨씬 적은 네트워크 부하를 가지게 되는 장점이 있다. 본 문서에서는 리눅스 기반의 ECTP-3 프로토콜을 이용한 간단한 응용프로그램을 실행시켜 보고 정상적으로 동작하는지 살펴본다.

2. ECTP 개요

ECTP(Enhanced Communications Transport Protocol)는 멀티캐스트를 지원하는 IP망을 바탕으로 한 인터넷 멀티캐스트 응용을 지원하기 위해 고안된 프로토콜이다. ECTP는 IGMP와 멀티캐스트 라우팅 프로토콜의 도움으로 멀티캐스트 전달능력이 있는 IPv4 또는 IPv6 망에서 동작한다. 또한 UDP 위에서도 동작이 가능하다. ECTP는 단방향과 양방향, N방향의 멀티캐스트 연결을 지원하기 위해서 설계되었다. ECTP는 아래의 6부분으로 나누어진다.

- ECTP-1 : Simplex multicast transport
- ECTP-2 : QoS management for Simplex multicast transport
- ECTP-3 : Duplex multicast transport
- ECTP-4 : QoS management for Duplex multicast transport
- ECTP-5 : N-plex multicast transport
- ECTP-6 : QoS management for N-plex multicast transport

2.1 ECTP-3 related Terms

ECTP-3에서 사용하는 용어는 많지만 이 문서에서 사용되는 용어는 아래와 같다.

- TO (TC-Owner)

TO는 ECTP-3 프로토콜의 전체 동작을 관리하며, TS-User들에게 멀티캐스트 데이터를 전송할 수 있다.

- TU (TS-User)

TU는 TO가 보낸 멀티캐스트 데이터를 받을 수 있다.

○ SU (Sending TU)

TO로부터 토큰을 받은 TU. SU만이 유니캐스트 데이터를 TO에게 전송할 수 있다. 즉, 유니캐스트 데이터 전송을 원하는 TU는 반드시 TO에게 토큰을 요청해야한다.

○ 토큰 (Token)

TU가 데이터를 전송할 수 있는 권리는 나타낸다. 토큰을 받은 TU를 SU라 부른다. 토큰을 TC-Owner에 의해 관리된다.

○ 순방향 데이터 채널 (Forward Data Channel)

TO로부터 TU로 전송되는 멀티캐스트 데이터 채널을 나타낸다. TO는 멀티캐스트 IP 주소상의 모든 그룹원에게 멀티캐스트 데이터를 전송한다.

○ 역방향 데이터 채널 (Backward Data Channel)

TU(SU)로부터 TO로 전송되는 유니캐스트 데이터 채널을 나타낸다. 토큰을 받은 SU는 유니캐스트 IP 주소상의 TO에게 유니캐스트 데이터를 전송할 수 있다.

2.2 ECTP-3 프로토콜 개요

ECTP-3는 ECTP 중에서 하나의 TC-Owner(TO)와 여러 TS-user(TU)로 구성되는 양방향 멀티캐스트 전송 연결을 위한 프로토콜 메커니즘이다.

ECTP-3에서는 다음과 같은 두 가지 형태의 데이터 전송을 지원한다.

○ 하나의 TO로부터 여러 TU로의 멀티캐스트 데이터 전송

○ 토큰을 획득한 여러 TU로부터 하나의 TO로의 유니캐스트 데이터 전송

ECTP-3의 연결 생성 단계는 TO에 의해 시작된다. 그런 다음 연결에 참여를 원하는 TU들이 TO에게 응답할 것이다. 연결 생성에서 확장성 있는 신뢰성 제어를 위해 제어 트리가 구성된다. 어떤 TU들은 낮은 가입 동작을 통해 참여할 수 있고, 어떤 TU들은 사용자 탈퇴 동작을 통해 연결에서 탈퇴한다.

연결이 설정된 후에는 데이터 전송 단계에 들어간다. TO는 그룹에게 멀티캐스트 데이터 전송을 시작할 것이다. 확장성 있고 신뢰성 있는 멀티캐스트 데이터의 전송을 제공하기 위해, TO와 모든 TU들 사이에는 논리적인 제어 트리가 구성된다. TO는 이러한 제어 트리의 근원지이다. 제어 트리는 부모-자식 관계를 정의한다. 그리고 어떤 TU들은 유니캐스트 데이터를 TO에게 전송할 수 있다. 데이터 전송을 원하는 TU는 먼저 TO로부터 토큰을 얻어야 한다. 토큰은 TO에게 유니캐스트 데이터를 전송할 수 있는 권리를 의미한다. 토큰을 가진 TU만이 TO로의 유니캐스트 전송이 허락된다.

데이터 전송 단계에서 멀티캐스트와 유니캐스트 데이터 전송이 완료되면, TO는 양방향 연결을 종료한다.

ECTP-3에서 양방향 멀티캐스트 연결은 TO가 모든 TU들에게 멀티캐스트 데이터 정보를 전송하고, TU들 중 토큰을 획득한 Send-user(SU)가 TO에게 유니캐스트 데이터를 전송하는 멀티캐스트 응용을 목표로 한다. 기본적으로 양방향 멀티캐스트 전송은 몇몇 TU들로부터의 유니캐스트 채널이 필요한 일대다 멀티캐스트 응용(원격 교육, 인터넷 방송 등)에 적합할 것이다. 예를 들면, 원격 교육 응용에서 강의자(TO)는 음성과 텍스트, 이미지와 같은 데이터를 학생(TU) 그룹에서 전송하고, 학생들 중 몇 명은 확인을 위해 질문과 같은 유니캐스트 데이터로 강의자에게 응답할 수 있다.

3. ECTP-3 APIs

ECTP-3의 API 함수는 3개의 부분으로 나눌 수 있다. 아래처럼 TO와 TU에서 모두 사용되는 API 함수와 TO에서만 사용되는 API 함수, TU에서만 사용되는 API 함수이다.

1) TO와 TU에서 사용되는 API 함수

○ msocket

ECTP-3을 위한 소켓 디스크립터를 생성하는 함수이다. 가장 처음 호출해야 하는 API 함수이다. 구현한 ECTP-3은 UDP 상에서 동작하므로 실질적으로 UDP 소켓이 생성된다.

○ mleave

연결 생성 과정을 거쳐 데이터 전송 과정에 있을 때 가입한 그룹을 탈퇴할 때 사용되는 API 함수이다. TO에서 사용하면 문제를 일으키는 TU를 추방하고 TU에서 사용하면 그룹을 탈퇴하는 동작을 수행한다. 구현한 ECTP-3에서는 TU가 그룹 탈퇴하는 동작만 수행한다.

○ mrecvmsg

ECTP-3에서 전송하는 데이터를 수신받는 API 함수이다. 멀티캐스트 데이터와 유니캐스트 데이터를 모두 받는다.

○ msetsockopt

ECTP-3의 소켓 옵션을 설정하는 API 함수이다. TCO, AGN, MSS, FDN의 값을 설정할 수 있다. 그리고 TO에서 가입 과정과 토큰 요청 과정에 호출되는 콜백 함수를 설정할 수 있다. 구현한 ECTP-3에서는 콜백 함수 설정을 제외한 설정은 임의적이지만 TO에서의 콜백 함수 설정은 이 API 함수를 이용하여 mcreate API 함수 호출 전에 반드시 수행해야 한다.

○ mgetsockopt

ECTP-3의 소켓 옵션을 확인하는 API 함수이다. TCO, AGN, MSS, FDN의 값을 확인할 수 있다.

2) TO에서만 사용되는 API 함수

○ mcreate

msocket API 호출로 소켓을 생성하고 msetsockopt 호출로 콜백 함수를 설정한 후 호출되는 이 API 함수는 TO의 연결 생성 과정을 담당한다. 이 함수를 호출하면 TO에서 TU를 가입시킬 때 필요한 값을 설정하고 CCT 타이머를 동작시킨다. 구현한 ECTP-3에서는 바로 가입하는 경우가 없기 때문에 CCT 타이머를 동작시키지 않고 바로 종료된다.

○ msendm

TO에서 데이터 전송 과정에 진입했을 때 순방향 멀티캐스트 데이터 채널을 통해 멀티캐스트 데이터를 전송하는데 사용하는 API 함수이다. 설정된 그룹 주소로 멀티캐스트 데이터를 전송한다.

○ mterminate

TO에서 멀티캐스트 데이터 전송을 마치고 종료할 때 호출하는 API 함수이다. 이것을 호출하면 순방향 멀티캐스트 데이터 채널을 통해 종료 메시지를 전송하고 msocket으로 생성한 소켓을 소멸시킨다.

○ mjoin_confirm

TU에서 늦은 가입을 요청할 때 수락 또는 거부할 것인지를 TO가 TU에게 알려주는 API 함수이다. 수락 또는 거부는 이 함수의 flag 값에 의한다.

○ mtoken_confirm

TU에서 유니캐스트 데이터를 TO에게 전송하기 위해 토큰을 요청할 때 수락 또는 거부할 것인지를 TO가 TU에게 알려주는 API 함수이다. 수락 또는 거부는 이 함수의 flag 값에 의한다.

3) TU에서만 사용되는 API 함수

○ mjoin

TU에서 TO에 가입 요청을 할 때 사용되는 API 함수이다. TO에서의 mcreate 함수와 같이 TU에서 필요한 값을 설정하고 TO에게 가입 요청을 한다. 구현한 ECTP-3에서는 늦은 가입의 경우만 고려했기 때문에 CWT 타이머를 동작시키지 않고 늦은 가입 과정으로 진입한다

○ msendmsg

TU에서 TO로 유니캐스트 데이터를 전송할 때 사용하는 API 함수이다. mtoken_get 함수를 호출하여 토큰을 획득한 경우 이 함수를 호출하여 역방향 데이터 채널을 통해 TO

에게 유니캐스트 데이터를 전송한다.

○ `mtoken_get`

TU가 TO에게 토큰을 요청할 때 사용하는 API 함수이다.

○ `mtoken_return`

TU가 TO에게 토큰 반환을 요청할 때 사용하는 API 함수이다.

4. 구현코드 및 설치

본 테스트 환경은 리눅스 기반으로 하였다. 리눅스 중에서도 이번 테스트에서 사용된 것은 커널 버전이 2.6.9인 페도라 3이다. 따라서 본 문서에서 테스트 환경은 페도라 3에서 수행하는 것을 바탕으로 설명할 것이고 Debian, 우분투 등 다른 리눅스를 사용할 경우 설치 및 설정 방법이 달라질 수 있다.

4.1 Library for Test Applications

4.1.1 Mplayer

Mplayer는 TU에서 멀티캐스트 채널로 수신받는 데이터를 플레이할 때 사용된다. Mplayer를 설치하는 방법에는 여러가지가 있다. 아래는 그 중 한가지 방법을 제시한다.

1) FC3 저장소 설정 파일을 설치한다.

```
http://ftp.freshrpms.net/pub/freshrpms/fedora/linux/3/freshrpms-release/ 경로에서
```

```
freshrpms-release-1-1.fc.fr.noarch.rpm 패키지를 다운받아서 설치한다.
```

```
]# rpm -Uvh freshrpms-release-1-1.fc.fr.noarch.rpm
```

2) GPG Key 설치

페도라 3의 yum 명령어를 사용하기 위해서 아래 명령을 수행해야 한다.

```
]# rpm --import /usr/share/doc/freshrpms-release-1/RPM-GPG-KEY-freshrpms
```

3) Mplayer 패키지 설치

```
]# yum install mplayer
```

4) Codec 설치

```
ftp://ftp5.mplayerhq.hu/MPlayer/releases/codecs/all-20061022.tar.bz2 에서 다운로드 받아
```

```
/usr/lib/win32 디렉토리에 아래처럼 압축을 푼다.
```

```
]# tar -xvjf all-20061022.tar.bz2
```

4.1.2 Qt Library

Qt Library는 응용 프로그램을 개발하는데 있어 GUI 부분을 위해서 필요하다. 이번 테스트에서 사용한 Qt Library의 버전은 3.3이나 최신 버전은 4.2이다. 페도라 3에서는 리눅스를 설치할 때 개발용 도구의 KDE 소프트웨어 개발을 선택하면 설치가 되어 있다. 만약 설치가 되어 있지 않다면 아래의 방법으로 설치하면 된다.

1) Qt Library 파일 다운로드

<http://www.trolltech.com/developer/downloads/qt/x11> 에서 다운로드 받는다

2) Qt Library 설치

```
]# tar -xvzf qt-x11-opensource-src-4.2.2.tar.gz  
]  
]# ./configure  
]  
]# make  
]  
]# make install
```

4.2 ECTP-3 Source Compile

4.2.1 File Structures

파일 구조는 3부분으로 나눌 수 있다. ECTP-3 프로토콜을 위한 헤더파일과 소스파일, 그리고 응용프로그램을 위한 파일이다. 파일 구조는 다음과 같다.

1) include 디렉토리

○ ectp.h

ECTP-3의 응용프로그램을 개발할 때 include 하는 기본헤더이다.

2) include/ectp 디렉토리

○ ectp.h

ECTP-3의 Core 부분에서 사용되는 함수를 정의한다.

○ structs.h

ECTP-3의 소켓 구조체와 프로토콜 제어 블록 부분을 정의한다.

○ types.h

ECTP-3의 내부 함수에서 사용되는 타입을 정의한다.

○ protocol.h

ECTP-3의 패킷 헤더와 패킷에 대해 정의한다.

○ user.h

ECTP-3의 소켓 API를 지원하기 위한 구조체와 함수를 정의한다.

○ event.h

ECTP-3의 내부적 이벤트와 관련된 이벤트 핸들러와 콜백 함수를 정의한다.

○ wrapper.h

ECTP-3의 내부에서 사용되는 에러 함수를 정의한다.

3) lib 디렉토리

○ socket.c

ECTP-3의 소켓 API를 위한 내부적인 함수를 구현해 놓은 파일이다.

○ control.c

ECTP-3의 제어 함수를 구현해 놓은 파일이다.

○ event.c

ECTP-3의 이벤트 콜백 함수와 핸들러를 구현해 놓은 파일이다.

○ timer.c

ECTP-3의 내부에서 사용하는 타이머를 구현해 놓은 파일이다.

○ user.c

ECTP-3의 소켓 API를 구현해 놓은 파일이다.

4) app 디렉토리

○ app/gserv

ECTP-3를 이용한 간단한 TO의 응용프로그램에 관한 파일들의 디렉토리이다.

○ app/gclnt

ECTP-3를 이용한 간단한 TU의 응용프로그램에 관한 파일들의 디렉토리이다.

4.2.2 Compile

ECTP-3 프로토콜의 컴파일은 gcc와 g++ 두 컴파일러 모두 사용이 가능하다. 본 테스트에서는 GUI를 위한 Qt Library가 C++ 기반이기 때문에 g++ 컴파일러를 이용하였다. 만일 GUI를 C API를 이용하여 작성한다면 gcc 컴파일러를 이용해도 된다.

첨부파일 ectp3.tar.bz2를 다운받아 압축을 푼다. ECTP-3의 컴파일은 build를 이용하여 컴파일한다. build는 여러가지 옵션이 있는데 그 중에서 주로 쓰이는 옵션은 다음과 같다.

○]# ./build all

ECTP-3 프로토콜의 소스 파일과 테스트 응용 파일을 컴파일한다. 테스트 응용의 실행파일과 libectp.so 라이브러리 파일이 생성된다.

○]# ./build ectp

ECTP-3 프로토콜의 소스 파일을 컴파일한다. libectp.so 라이브러리 파일이 생성된다.

○]# ./build clean

ECTP-3 프로토콜의 소스 파일과 테스트 응용 파일을 컴파일해서 생긴 목적코드 파일과 실행파일을 삭제한다.

○]# ./build tag

cscope 명령어를 이용하여 파일을 분석하기 쉽도록 tags를 생성한다.

5. 실험 결과

5.1 Test Scenario

테스트 시나리오는 하나의 TO와 두개의 TU를 이용하여 동작시켜 본다. TO의 응용프로그램에 해당 필드를 채우고 TO 응용프로그램을 실행한다. TO의 IP는 TO 자신의 주소이고 Group IP는 멀티캐스트를 전송할 그룹 주소이다. 두개의 TU는 모두 낮은 가입으로 TO에 연결을 생성하고 데이터를 받게 된다. TU는 멀티캐스트 데이터를 수신하면서 TO에게 데이터를 보내고 싶으면 토큰 요청 과정을 거쳐 TO가 수락하여 토큰 아이디를 부여하면 TO에게 유니캐스트로 데이터를 전송한다. 유니캐스트 전송을 마치면 토큰 반환 요청을 하여 토큰을 TO에게 되돌려준다. TO가 준비된 멀티캐스트 데이터를 모두 전송하면 종료하도록 한다.

주의 - TO 응용에서 멀티캐스트 데이터로 전송하는 동영상은 일반적인 포맷으로는 동작하지 않는다. TU는 낮은 가입 요청으로 TO에 가입을 하게 되어 파일의 중간부터 수신받는다. 따라서 중간부터 수신을 받아도 재생에 문제가 발생하지 않는 동영상 포맷을 사용해야 한다.

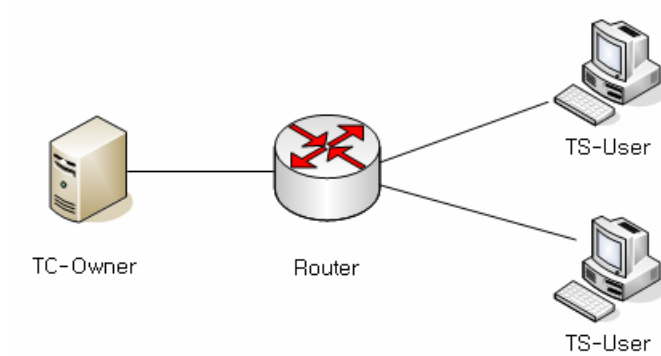


그림 1. 테스트베드

5.2 Results

아래의 화면은 실행한 응용프로그램의 캡처 화면이다. 왼쪽의 그림은 TO의 화면이고 오른쪽의 그림은 두개의 TU 중 하나의 화면이다. TO의 Local IP는 155.230.24.148이고 Group IP는 226.0.0.1, Group Port는 9926이다. TO는 위와 같이 설정하고 멀티캐스트 데이터 전송을 시작한다. 그 후 TU는 낮은 가입을 수행하고 멀티캐스트 데이터를 수신받는다. TU는 Mplayer로 수신받은 데이터를 보여주고 있는 것을 볼 수 있다. 그리고 TU는 토큰을 얻어 TO에게 'token data', 'good transmit' 라는 메시지를 전송한 것을 TO에서 확인할 수 있다.

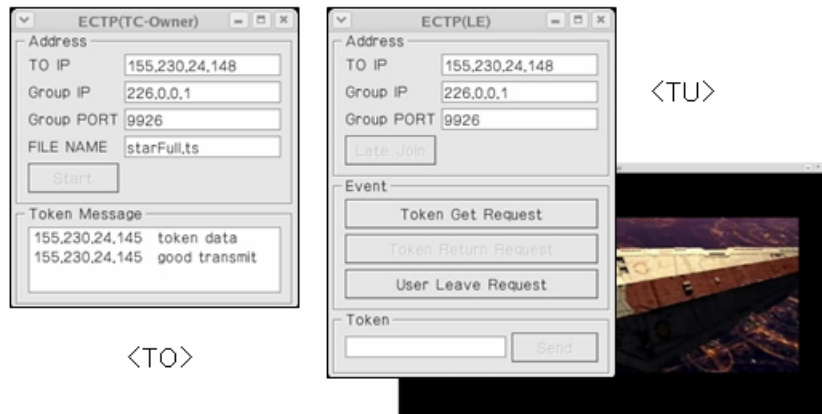


그림 2. 응용프로그램 실행 화면

6. 결론

이 문서에서는 ECTP-3의 소개와 테스트 해보기 위한 준비작업, 구현한 응용에 대한 실행 및 결과에 대해서 살펴보았다. 이번에 구현한 ECTP-3 프로토콜은 비신뢰, IPv4 기반으로 구현되어 있다. 이것을 신뢰적 기반과 IPv6도 지원할 수 있도록 보완하는 작업이 필요하다.

Reference

- [1] "Specification of duplex multicast transport", ITU-T X.607 | ISO/IEC 14476-3, ECTP-3
- [2] Media Player, <http://www.mplayerhq.hu/>
- [3] Qt Library for GUI, <http://www.trolltech.com>