

MMCP: Implementation and Experimentations

2008년 5월

경북대학교 통신프로토콜연구실

박재성 (knucsid@gmail.com)

Abstract

본 문서는 MMCP (ITU-T X.mmc-2 | ISO/IEC WD 24793-2) 표준 규격에 의거한 프로토콜 구현 코드 및 테스트 응용프로그램 실험에 관한 기술 문서이다.

<첨부파일: mmcp.tar.bz2>

Table of Contents

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

1. 서론

최근 IPTV의 연구가 활발하게 이루어지고 있다. 하지만 무선에서의 멀티캐스트 전송을 제어하는 프로토콜은 없어, 멀티캐스트의 전송을 제어하고 이동성을 지원하기 위한 프로토콜이 필요하다. MMCP(Mobile Multicast Control Protocol)는 IPTV 응용 데이터의 IP 멀티캐스트 전송에 대하여, 세션인증 제어, 과금을 위한 세션 상태감시 및 단말의 이동으로 인한 핸드오버 지원 기능을 제공하는 프로토콜이다. 이 프로토콜은 기존의 멀티캐스트 응용 프로그램의 기본적인 구조를 변경하지 않고 API를 통하여 적용할 수 있는 장점을 가지고 있다. 본 문서에서는 리눅스 기반의 MMCP 프로토콜을 이용한 간단한 응용 프로그램을 실행시켜보고 정상적으로 동작하는지 살펴본다.

2. MMCP 개요

MMCP(Mobile Multicast Control Protocol)는 멀티캐스트를 지원하는 무선 IP망을 바탕으로 비신뢰적인 실시간 멀티캐스트 응용을 타겟으로 한 프로토콜이다. MMCP는 IGMP/MLD와 멀티캐스트 라우팅 프로토콜의 도움으로 멀티캐스트 전달 능력이 있는 IPv4 또는 IPv6 망에서 동작한다.

2.1 MMCP Related Terms

MMCP에서 사용하는 용어는 많지만 이 문서에서 사용되는 용어는 아래와 같다.

- Session Manager (SM)
- Local Mobility Controller (LMC)
- Mobile Node (MN)
- Multicast Contents Server (MCS)
- Multicast Router (MR)
- Access Point (AP)

2.2 MMCP 프로토콜 개요

MMCP는 IP 멀티캐스트를 지원하는 네트워크에서 이동성을 제공하는 프로토콜이다. MMCP는 데이터 채널과 제어 채널이 분리되어 있어 데이터 채널은 관여하지 않고 멀티캐스트를 위한 제어 채널만 관여한다. 그림1은 MMCP의 일반적인 구조도를 보여주고 있다. 그림에서 보는 것과 같이 채널이 분리되어 있다.

MMCP는 크게 4가지 개체로 이루어진다. 일반적인 멀티캐스트 데이터를 전송하는 MCS와 인

증, 과금 등 전체적인 제어를 하는 SM이 있다. 또한 효율적인 상태 감시와 핸드오버 지원을 위해 각 서브넷마다 하나의 LMC가 존재하고 실질적인 서비스를 제공받는 사용자의 이동 단말인 MN이 있다.

MMCP은 3가지 중요한 동작을 한다. 그 3가지 동작은 인증 과정을 거친 세션 가입과 과금을 위한 상태감시, 이동성을 위한 지원이다. 세션 가입 과정은 MN에서 SM으로 가입 요청 메시지를 보내면 SM을 인증 과정을 거쳐 MN의 세션 가입을 허락하고 MN은 상태 감시 및 이동성 지원을 위해 LMC에 가입 요청을 한다. 가입이 완료되면 MN은 MCS로부터 멀티캐스트 데이터를 수신할 수 있고 주기적으로 LMC에게 상태 정보를 전송한다. LMC에서도 주기적으로 수신받은 MN들의 상태 정보를 통합하여 SM에게 전달한다. 그러다가 MN이 다른 서브넷으로 이동할 경우 MMCP의 핸드오버 과정이 수행되고 최소한의 멀티캐스트 데이터 끊김을 위해 이동할 LMC에서 미리 IGMP/MLD를 이용하여 멀티캐스트 데이터가 수신될 수 있도록 한다.

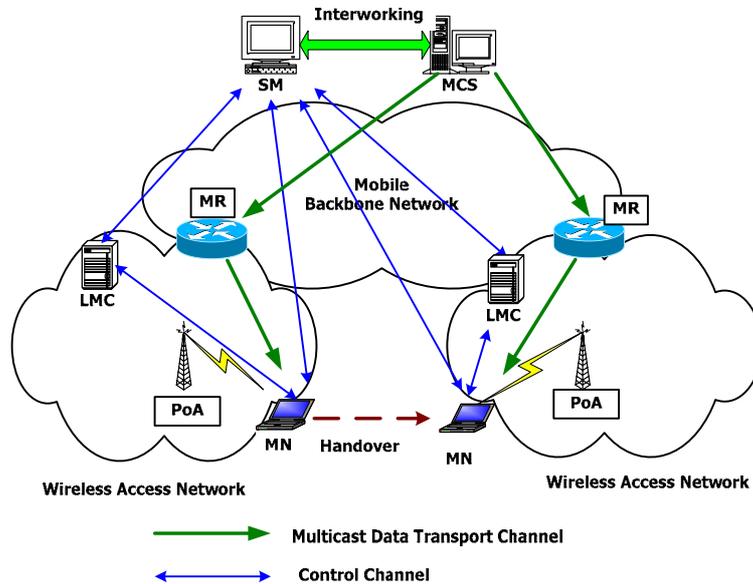


그림 1. 무선 네트워크 상에서의 MMCP 구조도

3. MMCP APIs

MMCP의 API 함수는 8개로 이루어져 있다. 각 API는 다음과 같다.

- mmcp_socket ()

MMCP를 위한 소켓 디스크립터를 생성한다. 이 소켓은 UDP 기반의 소켓이다. MMCP 개체에서 공통적으로 가장 처음 호출되는 API이며 mmcp_socket 구조체를 생성하고 초기화한다.

○ mmcp_create()

Session Manager에 의해서 호출되는 API로써 mmcp_endpoint 구조체와 mmcp_sm 구조체를 생성하고 초기화한다. SM을 위한 함수이며 멀티캐스트 데이터 채널을 위한 소켓을 저장하고 제어 채널을 위한 소켓을 바인딩한다.

○ mmcp_processing()

이 함수는 SM에서 호출하는 API로써 MMCP 프로토콜을 수행하기 위한 main_loop를 동작시키고 세션 가입 상태 감시를 위한 이벤트 스레드를 생성 및 동작시킨다. 이 함수는 IDLE 함수이다.

○ mmcp_local_join()

Local Mobility Controller에 의해서 호출되는 API로써 mmcp_endpoint 구조체와 mmcp_lmc 구조체를 생성하고 초기화한다. LMC를 위한 함수이며 LMC가 SM에 세션 가입하는 과정을 수행한다.

○ mmcp_join()

Mobile Node에 의해 호출되는 API로써 mmcp_endpoint 구조체와 mmcp_mn 구조체를 생성하고 초기화한다. MN을 위한 함수이며 MN이 세션 가입에 필요한 절차를 수행한다.

○ mmcp_leave()

현재 유지하고 있는 MMCP 세션을 종료하고자 할 때 사용하는 API이다. 생성된 구조체와 사용하고 있는 리소스를 해제한다.

○ mmcp_setsockopt()

MMCP의 내부적인 매개변수를 설정할 때 사용하는 API이다. 설정할 수 있는 매개변수는 일반적으로 타이머가 있다.

○ mmcp_getsockopt()

MMCP의 내부적인 매개변수를 확인할 때 사용하는 API이다. 확인할 수 있는 매개변수는 일반적으로 타이머가 있다.

4. 구현코드 및 설치

본 테스트 환경은 리눅스 기반으로 하였다. 리눅스 중에서도 이번 테스트에서 사용된 것은 커널 버전이 2.6.20인 우분투 7.04이다. 따라서 본 문서에서 테스트 환경은 우분투 7.04에서 수행하는 것을 바탕으로 설명할 것이고 페도라, 데비안 등 다른 리눅스를 사용할 경우 설치 및 설정 방법이 달라질 수 있다.

4.1 Library for Test Applications

본 실험에서 사용되는 라이브러리는 `libpcap`과 `mplayer`이다. 그것의 설치와 쓰임새를 알아보도록 한다.

4.1.1 Libpcap

`Libpcap`은 이동성을 지원하기 위해 사용되는 라이브러리이다. 이동성을 지원하기 위해 다른 무선 신호를 감지해야 하는데 그것을 감지하고 구분해주는 라이브러리이다. 각각의 리눅스 플랫폼마다 최신 버전의 `libpcap`을 설치하면 된다. 본 실험에서 설치되어 있는 버전은 우분투에서의 버전이다.

```
l$ sudo apt-get install libpcap0.8
l$ sudo apt-get install libpcap0.8-dev
```

4.1.2 Mplayer

`Mplayer`는 MN에서 수신받은 멀티캐스트 데이터를 동영상 플레이할 때 사용된다. `Mplayer` 이외에도 다른 동영상 플레이어가 많이 있으므로 다른 프로그램을 사용해도 무방하다. 본 MN 구현에서 `Mplayer`를 사용하였기 때문에 예제로 제시한다.

`Mplayer`를 설치하는 방법에는 여러 가지가 있으나 우분투에서는 비교적 간단하게 설치할 수 있다.

```
l$ sudo apt-get install mplayer
```

4.2 MMCP Source Compile

4.2.1 File Structures

파일 구조는 4부분으로 구성되어 있다. MMCP 프로토콜을 위한 헤더파일과 소스파일, 그리고 응용 프로그램을 위한 파일이다. 그 외에 본 소스 파일을 컴파일 하기 위한 `Make` 파일과 `Build` 파일 및 프로토콜 동작시 필요한 `shell` 파일이 존재한다. 4부분으로 구성된 파일 구조는 다음과 같다.

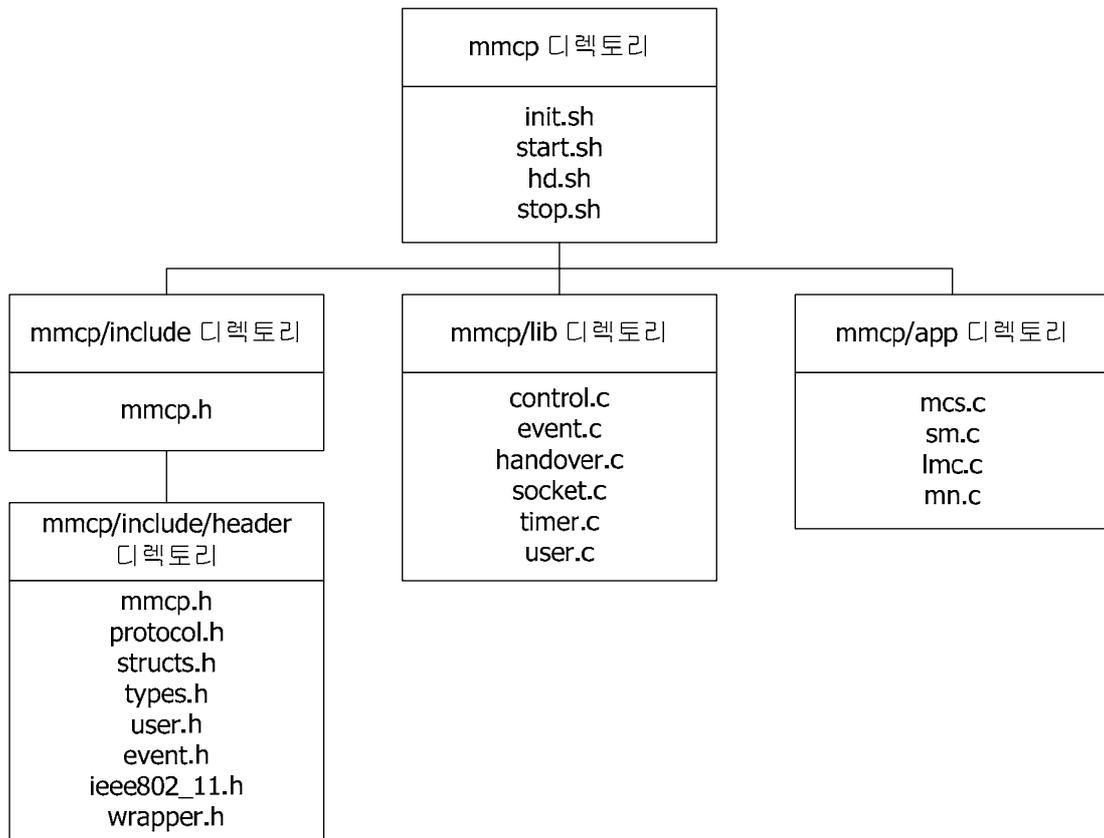


그림 2. 파일 구성도

4.2.2 Compile

MMCP 프로토콜의 컴파일은 gcc와 g++ 두 컴파일러 모두 사용이 가능하다. 본 테스트에서는 응용 프로그램을 C 기반으로 작성하였기 때문에 gcc 컴파일러를 사용하였으나 만일 응용 프로그램을 C++ 기반으로 작성하였다면 g++ 컴파일러는 사용하면 된다. 본 테스트에서의 컴파일러 버전은 4.1.2이다.

첨부파일 mmcp.tar.bz2 파일을 다운받아 압축을 푼다. 압축을 푸는 방법은 다음과 같다.

```
l$ tar -xvjf mmcp.tar.bz2
```

압축을 푼 후 MMCP의 컴파일은 build를 이용하여 컴파일 한다. build에는 여러 가지 옵션이 있는데 그 중에서 주로 사용되는 옵션은 다음과 같다.

```
l$ ./build all
```

MMCP 프로토콜의 소스 파일과 테스트 응용 파일을 컴파일한다. 테스트 응용의 실행 파일과 libmmcp.so 라이브러리 파일이 생성된다.

```
]$ ./build mmcp
```

MMCP 프로토콜의 소스 파일만 컴파일한다. libmmcp.so 라이브러리 파일이 생성된다.

```
]$ ./build clean
```

MMCP 프로토콜의 소스 파일과 테스트 응용 파일을 컴파일시 생성된 목적코드 파일과 실행 파일 및 라이브러리 파일을 삭제한다.

```
]$ ./build tag
```

cscope 명령어를 이용하여 파일을 분석하기 쉽도록 tags를 생성한다.

```
]$ ./build backup
```

MMCP 프로토콜의 소스 파일을 압축하여 mmcp-yyyymmdd-tt.tar.bz2 형태의 파일을 생성한다.

5. 실행 결과

5.1 Test Scenario

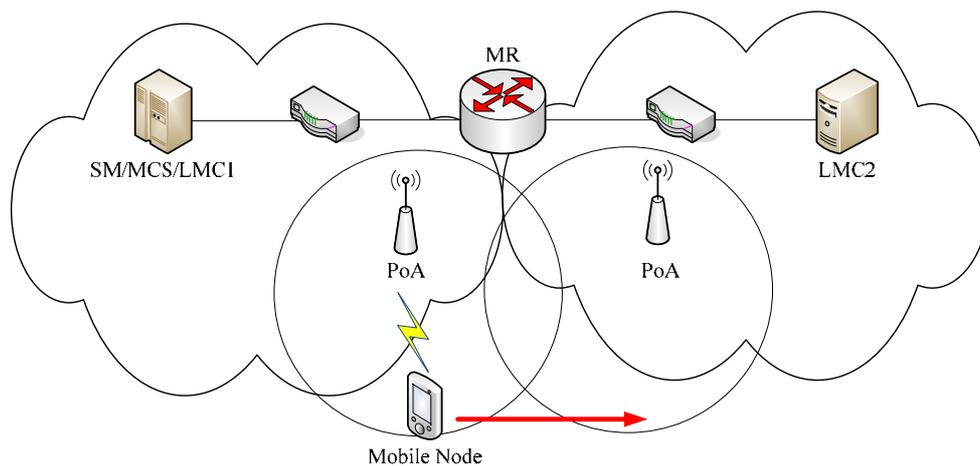


그림 3. 테스트베드

그림 1은 테스트베드의 구성을 나타내고 있다. 테스트 시나리오는 하나의 SM과 MCS가 있

고 각 서브넷에 LMC를 하나씩 배치한다. 이렇게 배치된 상태에서 MN이 서브넷을 이동할 때 프로토콜의 동작 상태를 살펴보도록 한다. 최초 MCS가 동작되고 그 후 SM을 동작시킨다. 그리고 LMC1과 LMC2를 차례로 SM에 가입시킨 후 MN을 LMC1에 세션 가입시킨다. 그러면 MN은 MCS로부터 멀티캐스트 데이터를 수신 받게 된다. 그 후 다른 서브넷으로 이동하게 되면 핸드오버 과정을 수행하여 LMC2로 이동한 후 멀티캐스트 데이터를 받도록 한다.

각 응용 프로그램의 실행 방법은 다음과 같다.

○ Multicast Contents Server

```
]$ ./rummcs [Group IP] [Group Port] [Filename]
ex) ]$ ./rummcs 226.0.0.1 9999 TestVod.ts
```

○ Session Manager

```
]$ ./rumsm [SM IP] [Group IP] [Group Port]
ex) ]$ ./rumsm 192.168.1.2 226.0.0.1 9999
```

○ Local Mobility Controller

```
]$ ./rumlmc [Local IP] [SM IP] [Local Port]
ex) ]$ ./rumlmc 192.168.1.3 192.168.1.2 9926
```

○ Mobile Node

MN을 동작시키기 전에 각 MN마다 특성이 다르므로 자신의 MN에 맞는 설정을 한 후 동작시켜야 한다.

본 구현에서는 인터페이스 카드를 2개 사용하여 하나는 실제 통신용으로 사용하고 다른 하나는 이동 감지를 위해서 사용된다. 따라서 실제 통신용으로 사용하는 인터페이스와 이동 감지를 위해 사용하는 인터페이스를 지정해주어야 한다.

```
]$ vi mmcp/include/header/mmcp.h
#define MMCP_CAPTURE_DEV [이동감지 인터페이스 이름]
#define MMCP_USE_EV      [통신용 인터페이스 이름]
```

또한 SHELL 파일도 MN에 맞게 변경해 주어야 한다. 이것은 본 실험에서 사용된 우분투에서의 설정 파일이므로 다른 플랫폼에서는 변경될 수도 있다.

```
]$ vi init.sh
iwconfig [통신용 인터페이스 이름] mode managed essid [초기 AP 이름]
]$ vi start.sh
iwconfig [이동감지 인터페이스 이름] mode monitor
]$ vi hd.sh
iwconfig [통신용 인터페이스 이름] mode managed essid [이동할 AP 이름]
dhclient [통신용 인터페이스 이름]
]$ vi stop.sh
iwconfig [이동감지 인터페이스 이름] mode managed
```

위의 설정을 모두 마치면 다음과 같이 MN을 동작시킨다.

```
]# ./rummn [SM IP]
ex) ]# ./rummn 192.168.1.2
```

- ※ 주의1 - MCS 응용에서 멀티캐스트 데이터로 전송하는 동영상은 일반적인 포맷으로는 동작하지 않는다. 그 이유는 MCS 응용이 동작하고 있는 가운데 MN이 세션 가입을 함으로써 동영상의 중간부터 수신 받게 된다. 따라서 중간부터 수신을 받아도 재생에 문제가 발생하지 않는 동영상 포맷을 사용해야만 한다.
- ※ 주의2 - MN의 경우 이동을 감지하기 위해서 PoA의 비콘을 감시하게 된다. 본 구현에서는 이 스캔을 위해 libpcap을 사용하는데 이 libpcap은 root 권한이 있어야 정상적으로 수행된다. 따라서 다른 응용은 관계가 없으나 MN 응용을 실행할 때에는 root 권한으로 동작시켜야만 한다.

5.2 Results

아래의 화면은 실행한 응용 프로그램의 캡처 화면이다. SM의 화면은 LMC로부터 통합된 MN의 정보를 보여주고 있고 LMC의 화면은 MN이 핸드오버를 요청했을 때의 그 처리 과정을 보여주고 있다. MN의 화면은 MN이 다른 서브넷으로 이동했을 때 주소를 재설정하고 해당 서브넷의 LMC에게 가입 절차를 보여주고 있다. MN은 MCS로부터 멀티캐스트 데이터를 수신하여 mplayer를 통해 동영상을 플레이할 수 있다.

<Session Manager>

```

knucsid@knucsid-sm:~/mmcp
File Edit View Terminal Tabs Help
knucsid@knucsid-sm:~/mmcp
MMCP, Receive SJR
SJR packet received(MN).
MN Identification : JS.Park
MN of PoA Identification : CPL-API
Session Join Confirm Message Send!

==== Aggregation Status Information of PoA ID: CPL-API
==== Aggregation Status Information of PoA ID: CPL-API
MN Identification: JS.Park
Total number of packets received: 1334094
Elapsed Time: 5

==== Aggregation Status Information of PoA ID: CPL-API
MN Identification: JS.Park
Total number of packets received: 3498408
Elapsed Time: 13

==== Aggregation Status Information of PoA ID: CPL-API
MN Identification: JS.Park
Total number of packets received: 5662644
Elapsed Time: 21
    
```

<Local Mobility Controller>

```

knucsid@knucsid-sm:~/mmcp
File Edit View Terminal Tabs Help
knucsid@knucsid-sm:~/mmcp
==== Status Information of MN ID: JS.Park ====
Total number of packets received (byte): 12673608
Elapsed Time: 32

==== Status Information of MN ID: JS.Park ====
Total number of packets received (byte): 16399173
Elapsed Time: 40

MMCP, Receive HIR Packet.
LMC receives Handover Initiation Request Packet.
MN Identification: JS.Park
MN of PoA Identification: CPL-API
MMCP, Waiting Handover Transfer Acknowledgment! (STATE: ESTABLISHED)
LMC receives Handover Transfer Acknowledgment Packet.
MMCP, Handover Transfer Completed! (STATE: ESTABLISHED)
Send HIC

==== Status Information of MN ID: JS.Park ====
Total number of packets received (byte): 18568673
Elapsed Time: 48
    
```

<Mobile Node>

```

knucsid@knucsid-mobile:~/mmcp-final-workings
File Edit View Terminal Tabs Help
MMCP, Handover Initiation Completed! (STATE: ESTABLISHED)
There is already a pid file /var/run/dhclient.pid with pid 12742
killed old client process, removed PID file
Internet Systems Consortium DHCP Client V3.0.4
Copyright 2004-2006 Internet Systems Consortium.
All rights reserved.
For info, please visit http://www.isc.org/sw/dhcp/

Listening on LPF/rausb0/00:17:31:eb:9c:c2
Sending on LPF/rausb0/00:17:31:eb:9c:c2
Sending on Socket/fallback
DHCPREQUEST on rausb0 to 255.255.255.255 port 67
DHCPREQUEST on rausb0 to 255.255.255.255 port 67
DHCPREQUEST on rausb0 to 255.255.255.255 port 67
DHCPDISCOVER on rausb0 to 255.255.255.255 port 67 interval 4
DHCPOFFER from 192.168.2.254
DHCPREQUEST on rausb0 to 255.255.255.255 port 67
DHCPACK from 192.168.2.254
bound to 192.168.2.4 -- renewal in 366196 seconds
MMCP, Waiting Local Join Confirm! (STATE: LJR-S)
MMCP, Local Join of Handover Confirm!
MMCP, Local Join Completed! (STATE: ESTABLISHED)

knucsid@knucsid-mobile:~/mmcp-final-workings
    
```



6. 결론

이 문서에서는 MMCP의 소개와 구현한 MMCP 프로토콜을 테스트 하기 위한 준비작업, 구현한 응용에 대한 실행 및 결과에 대해서 살펴보았다. 이번에 구현한 MMCP 프로토콜은 IPv4 기반으로 구현되어 있다. 향후 IPv6도 지원할 수 있도록 보완하는 작업이 필요하다.

Reference

- [1] Mobile Multicast Communications: Framework, ITU-T X.mmc-1 | ISO/IEC FCD 24793-1, mmc-1, April 2008
- [2] Mobile Multicast Communications: Protocol over Native IP Multicast Networks, ITU-T X.mmc-2 | ISO/IEC CD 24793-1, mmc-2, April 2008
- [3] Media player, <http://www.mplayerhq.hu/>
- [4] Tcpdump/Libpcap, <http://www.tcpdump.org>