

# Implementation of Mobile-Oriented Future Internet (MOFI) using Openflow Virtual Switch & Click Router

March 2014

Nak Jung Choi

School of Computer Science and Engineering

Kyungpook National University (KNU)

## 요 약

본 문서에서는 MOFI Testbed 구현을 하기 위해 Openflow Virtual Switch 와 Click Router를 이용한 구현에 대한 기술 문서이다.

## Table of Contents

1. 서론 .....	2
2. MOFI TESTBED 소개 .....	3
2.1 Host 세팅 및 설치방법 .....	4
2.2 AR 세팅 및 설치 과정.....	5
2.3 GATEWAY 세팅 및 설치 과정 .....	6
2.4 OPENSWITCH CONTROLLER .....	8
3. MOFI 구현 테스트 결과 .....	9
3.1 구현 TEST 환경.....	9
3.2 MOFI 실험 결과 .....	10
4. 결론 .....	11

## 1. 서론

미래 인터넷에서의 이동성을 우선으로 하는 Mobile Oriented Future Internet (MOFI) 구조는 기본적으로 다음 세 가지의 원칙을 가지고 디자인된 미래 인터넷 구조이다.

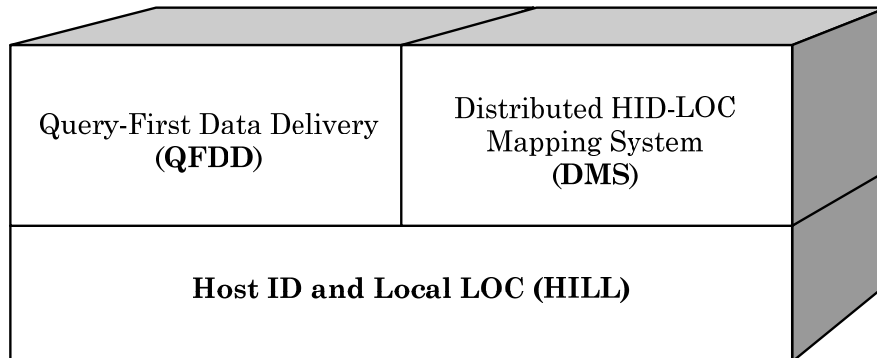


Figure 1. MOFI 구조

본 문서에서는 이러한 MOFI 구조를 현재 미래 인터넷 기술 개발에 많이 사용되고 있는 Openflow 기반으로 구현을 하기 위한 방법에 대하여 서술하겠다. 본 문서에서는 Openflow 기술 중 하나인 Openflow Virtual Switch(OVS)를 사용하여 구현을 하는 방법에 대하여 서술 하겠다.

또한 OVS 기술 만으로는 HILL 구조의 구현이 힘들기 때문에 HILL 구조 구현을 위한 Click Router 또한 사용할 것이다. Click Router는 응용 패킷의 encapsule과 decapsule을 통하여 원하는 Stack Layer를 추가 할 수 있다. 원래 MOFI Spec 문서 상의 HID는 128 Bit로서 정의 되어 있지만 구현의 편의를 위하여 HID를 현재 사용하고 있는 IPv4 헤더의 크기와 동일하게 가정하여 IPv4 를 하나의 HID로 만들고 그 패킷에 인캡슐화를 통해서 Local Locator 정보 또한 IPv4를 사용하도록 할 것이다.

또한 현재 응용 HID로는 IPv6를 응용하여 수정 후 통신을 하려는 모습을 생각하였지만 현재 Openflow 기반으로 통신 환경을 구축하기 위해서 좀더 구현에 용이한 IPv4 정보를 HID로 가정하여 사용하였다.

## 2. MOFI Testbed 소개

MOFI Testbed 환경 구축을 위해서 기본적인 배경 환경 세팅이 필요합니다. MOFI Testbed 환경을 구축하기 위해 기본적인 OS 설정을 리눅스 환경으로 구축을 해주어야 합니다. 그 이유는 현재 Testbed 개발을 위해서 MOFI 만의 고유의 Stack등일 필요하기 때문에 OpenSource로 구성되어 있는 리눅스 환경으로 구현을 하였습니다.

기본적으로 MOFI 구현을 위해서 사용한 OS 환경 및 개발 환경은 다음과 같다.

- OS : Ubuntu 12.04 LTS (32bit version)
- Openvswitch version 1.9.0
- Click Router

기본적인 OS는 Ubuntu 12.04 LTS로 동일안 Version 을 사용하였으나 OS 변경을 하게 되더라도 구현에 많은 문제가 일어나지는 않을 것으로 알고 있다. 하지만 OS를 변경하게 되면 각각의 기능을 구현하는 기본 프로그램 설치 과정에 있어서 명령어 변경이 있을 수도 있습니다. 아래 그림은 전체적인 현재 구현해둔 네트워크 상황을 정리한 Testbed 환경 모습입니다.

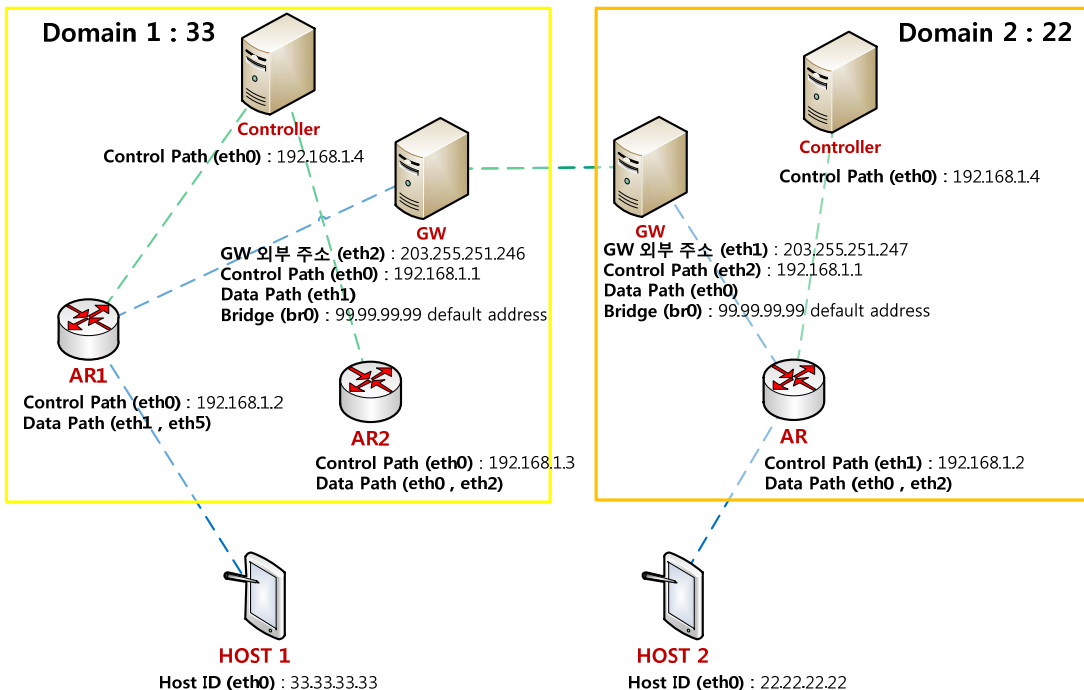


Figure 2. MOFI 구현 Testbed 환경

## 2.1 Host 세팅 및 설치방법

Host의 경우에는 별도의 처리 및 설치 과정 없이 기본적으로 사용하는 리눅스 PC를 사용하면 됩니다. Host의 경우에는 Ubuntu 12.04 LTS 버전을 그대로 설치를 하고 난 뒤에 그 위에 리눅스에서 사용되고 있는 응용 프로그램들을 사용하게 됩니다. 그리고 Host에서는 MOFI에서 사용되는 고유 ID가 필요하기 때문에 현재 사용되고 있는 IPv4 주소를 응용하여 사용해주어야 합니다. 그리고 그 고유 ID (Host ID)를 사용하기 위해서 등록을 위한 컨트롤 Message를 만들어 사용하였습니다. 그 message는 UDP 응용으로서 자신의 HID를 등록하기 위한 정보를 전달해주는 역할을 하게 됩니다. 원래의 경우에는 모든 컨트롤 message들이 독립적으로 MOFI에서는 응용 되지만 Openvswitch를 사용하게 되어 다음과 같이 붉은 선 부분을 Openvswitch 컨트롤 message를 그대로 응용하여 사용하였습니다.

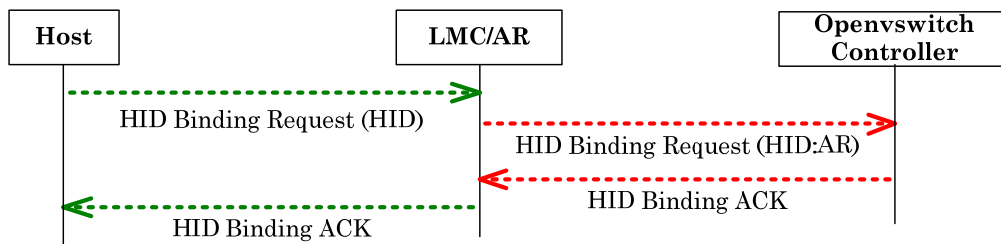


Figure 3. HID Binding 업데이트 과정

위와 같은 동작을 위한 세팅 방법은 아래와 같습니다.

- `ifconfig [네트워크 카드 이름] [Host ID]`
- `g++ reg.c -o reg && ./reg [Host ID]`

Figure 2 기준으로 조금더 세부 설명을 하게 되면 네트워크 카드 이름의 경우에는 eth0 가 되는 것이고 Host ID 의 경우에는 33.33.33.33으로 설정을 하게 됩니다. 그리고 reg.c 의 경우에는 구현된 코드로 따로 만든 소스 프로그램이다. 소스 코드 모두를 여기에 넣을 수는 없으므로 중요한 부분으로는 설정해둔 Host ID의 정보를 Controller가 설정되어 있는 곳으로 정보를 주어 등록 되는 과정이다. 등록하는 컨트롤 message는 row socket을 응용하여 구현하였다.

## 2.2 AR 세팅 및 설치 과정

AR의 경우에는 Openvswitch를 설치하고 실행을 해주게 되면 컨트롤 message를 통해서 등록과 데이터 전달 과정을 동작하게 된다. 그러므로 AR의 경우에도 기본적인 리눅스 OS를 설치해서 사용하면 된다. 그 후에 Openvswitch 설치를 하면 된다. 하지만 여기서 OS의 version에 따라서 설치하는 방법이 조금씩 다르다. 하지만 Openvswitch 버전은 동일하게 설치를 할 수 있다. 이 문서의 경우에는 Ubuntu LTS 12.04에서 설치하는 방법에 대하여 설명을 하겠다.

### - **apt-get install openvswitch-switch (Openvswitch 설치 과정)**

위와 같이 한줄의 명령어로 Openvswitch를 설치 할 수 있다. 하지만 Ubuntu가 아닌 다른 종류의 리눅스의 경우에는 설치과정이 조금 복잡할 수 있다. 각각의 리눅스 종류별로 설치 방법은 쉽게 알 수 있으므로 설치 과정에서는 실수만 없으면 설치가 가능할 것이다.

설치를 다 하고 난뒤에는 MOFI 동작을 지원하기 위한 기본적인 세팅이 필요하다. 그 세팅 과정은 다음과 같은 명령을 순서대로 입력 하면 되는 것이다. 이 세팅 방법에 사용된 명령들은 위에 나온 Figure 2에서 사용된 디바이스 명으로 설명을 하도록 하겠다.

### - **ovs-vsctl add-br bro && ifconfig bro 192.168.1.2/24 up**

(Openvswitch 컨트롤 message를 주고 받기 위한 가상 네트워크 구성후 IP 세팅)

### - **ovs-vsctl add-port bro eth1 && ovs-vsctl add-port bro eth5**

(Openvswitch 에서 data 전달 및 data 전송을 위한 port 연결 부분)

### - **ovs-vsctl set-controller bro tcp:192.168.1.4/6633**

### - **ovs-vsctl set controller bro connection-mode=out-of-band**

(Openvswitch 동작을 도와주는 Controller와 연결 해 주는 부분)

현재 구현되어 있는 MOFI의 AR이 가지는 주된 동작 처리 과정은 컨트롤 message전송을 통한 등록 과정 및 data path를 통한 data 전달 과정을 담당하게 된다. 자세한 message 종류는 GW 설명 부분에서 좀더 자세하게 설명을 하겠다.

### 2.3 Gateway 세팅 및 설치 과정

GW의 경우에는 AR과 마찬가지로 Openswitch 설치가 필요로 하다. 거기에 Click Router 설치가 추가로 필요로 하다. Click Router의 경우에는 intra-domain 통신에서는 사용이 되지 않는 것 이지만 inter-domain으로 바뀌었을 경우에는 location 변경을 위하여 사용되는 기술이다. 우선 intra-domain 경우의 통신하는 방법이다.

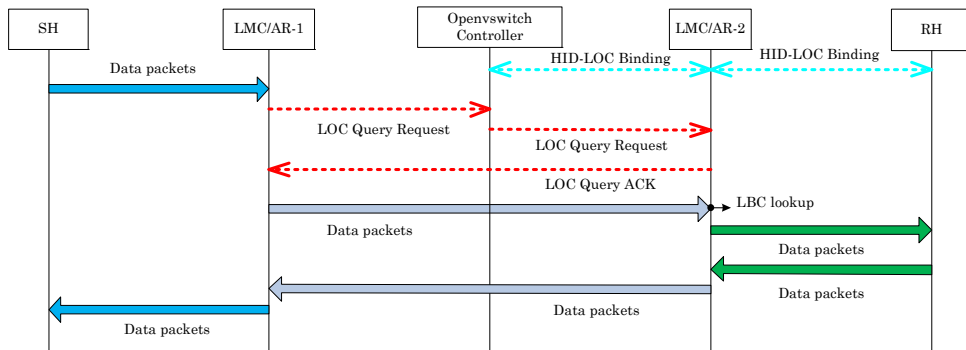


Figure 4. intra-domain 통신 순서

Figure 4를 보게 되면 역시 Openswitch 컨트롤 message를 이용하여 LOC Query를 응용한다. 즉 intra 통신의 경우에는 Openswitch를 응용하여 통신한다고 말할 수 있다. 위의 플로우에서 붉은 선이 원래 동작을 하기 위한 부분에서 Openswitch 컨트롤 message로 교체 되는 과정이다.

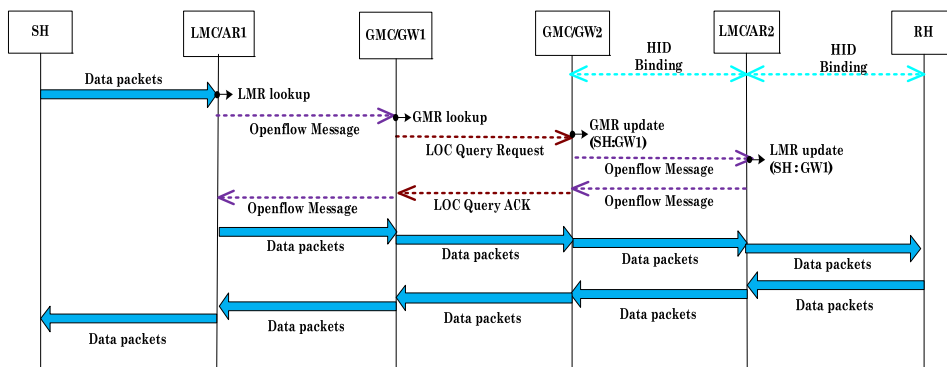


Figure 5. Inter-domain 통신 순서

Figure 5 의 경우에는 inter-domain 경우의 통신 이다. 우선적으로 inter-domain 경우에는 상대편 host가 같은 내부망인지 아닌지를 판별하여 통신해야 하는 것이 필요로 한다. 그렇기 때문에 그 과정을 위해서는 Host ID의 정보를 통해서 domain 정보를 파악하게 된다. 그 후 해당 domain에 위치 하고 있는지를 알아보는 컨트롤 message를 보내게 되고 그 후 그 domain 여부에 따라서 GW location 정보를 이용하여 data packet을 전송하게 된다. 이 역할을 수행해주는 부분이 Click Router를 통해서 하게 된다. 그 내부의 과정은 다음과 같다.

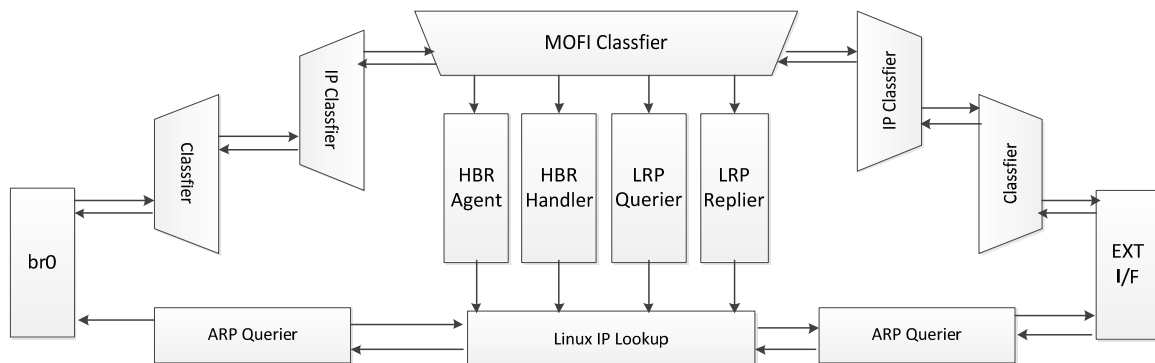


Figure 6. MOFI 동작을 위한 Click Router Process

각각의 부분을 설명을 하자면 MOFI Classifier에서는 크게 두가지를 구분하게 된다. Host ID 정보를 업데이트 하기 위한 Host Binding Request/Reply (HBR)을 업데이트 하는 과정 그리고 Location Register Process (LRP) 라고 하여 Gateway에서 Location 정보를 업데이트 하는 과정이다. Click Router의 내부 코드는 위에 동작과정을 반영하여 만들었다.

그럼 세팅을 위한 GW 설치 방법 및 세팅을 하기 위한 과정을 설명하겠다. 우선적으로 Click Router를 설치해야 한다.

- wget [www.read.cs.ucla.edu/click/click-2.0.1.tar.gz](http://www.read.cs.ucla.edu/click/click-2.0.1.tar.gz)
- cp MOFI directory click-2.0.1/elements/local
- ./configure --enable-local --disable-linuxmodule
- make elemlist && make

그리고 난 뒤에 메인적인 동작을 위한 Openvswitch 설치 과정과 세팅 과정이 필요하다.

- apt-get install openvswitch-common openvswitch-switch g++ make
- ovs-vsctl add-br bro && ifconfig bro 99.99.99.32 up
- ip route add default dev bro

- `ovs-vsctl add-port bro eth1 && ovs-vsctl add-port bro eth2`
- `ovs-vsctl set-controller bro tcp:192.168.1.4/6633`
- `ovs-vsctl set controller bro connection-mode=out-of-band`

이렇게 세팅을 하면 기본적인 동작을 위한 설치 및 세팅을 마무리 한 과정이다. 그리고 난 뒤에 만들어 놓은 Click Router 동작을 수행하게 된다.

- In `test.sh` fix, modify the `EXTERNAL DEVICE` symbol
- Run `./test.sh` to generate `mofi_test.click` script
- `userlevel/.click mofi_test.click` to start the router

### 2.4 Openswitch Controller

Openswitch Controller 부분 같은 경우에는 원래 기존의 MOFI spec에서는 필요로 하는 경우가 없다. 하지만 현재 구현을 위한 tool로 Openswitch를 사용하기 때문에 필요로 하는 부분이다. 그래서 간단한 설치 과정을 통한 세팅을 해주게 되면 동작을 하게 된다.

- `apt-get install openswitch-controller`
- `./openswitch-controller`



### 3. MOFI 구현 테스트 결과

#### 3.1 구현 Test 환경

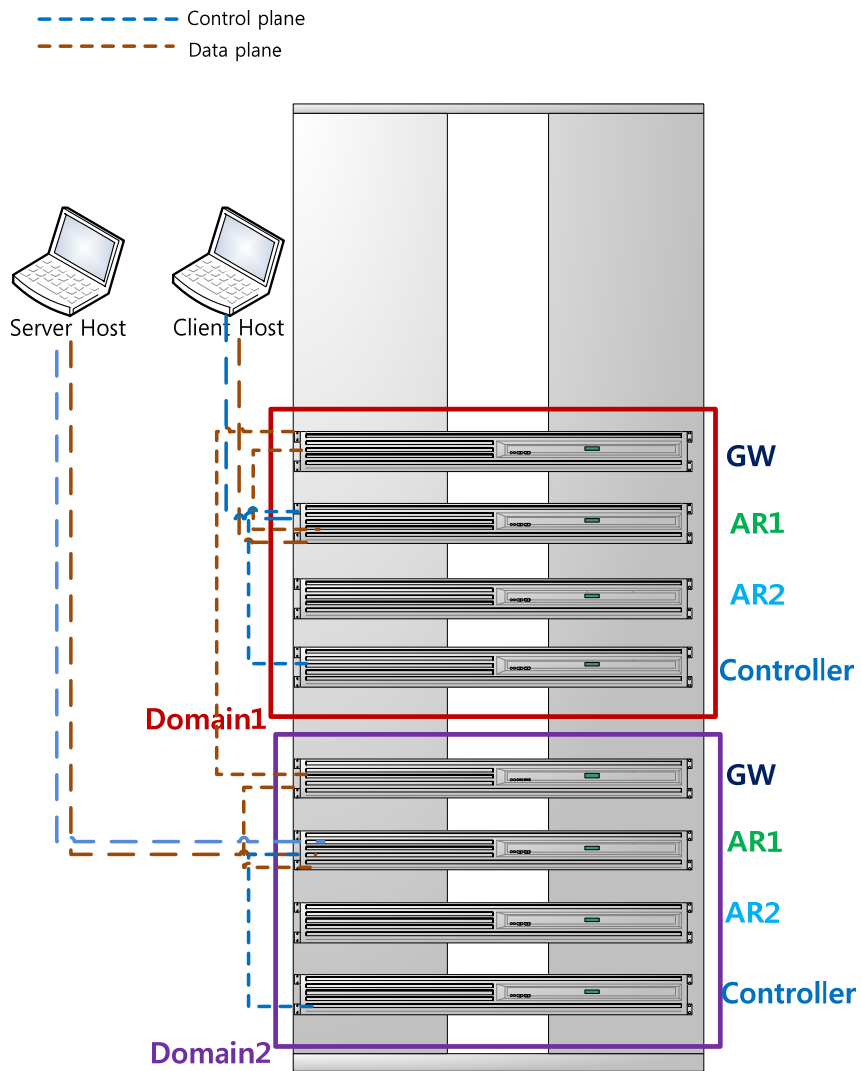


Figure 7. MOFI Testbed 구현 환경

.위의 그림과 같이 Testbed를 서버랙에 구성하여 실험을 하고 현재 서버를 구동하고 있다.  
현재 서버랙의 상황상 두개의 domain을 만들어서 동작을 하였다.

### 3.2 MOFI 실험 결과

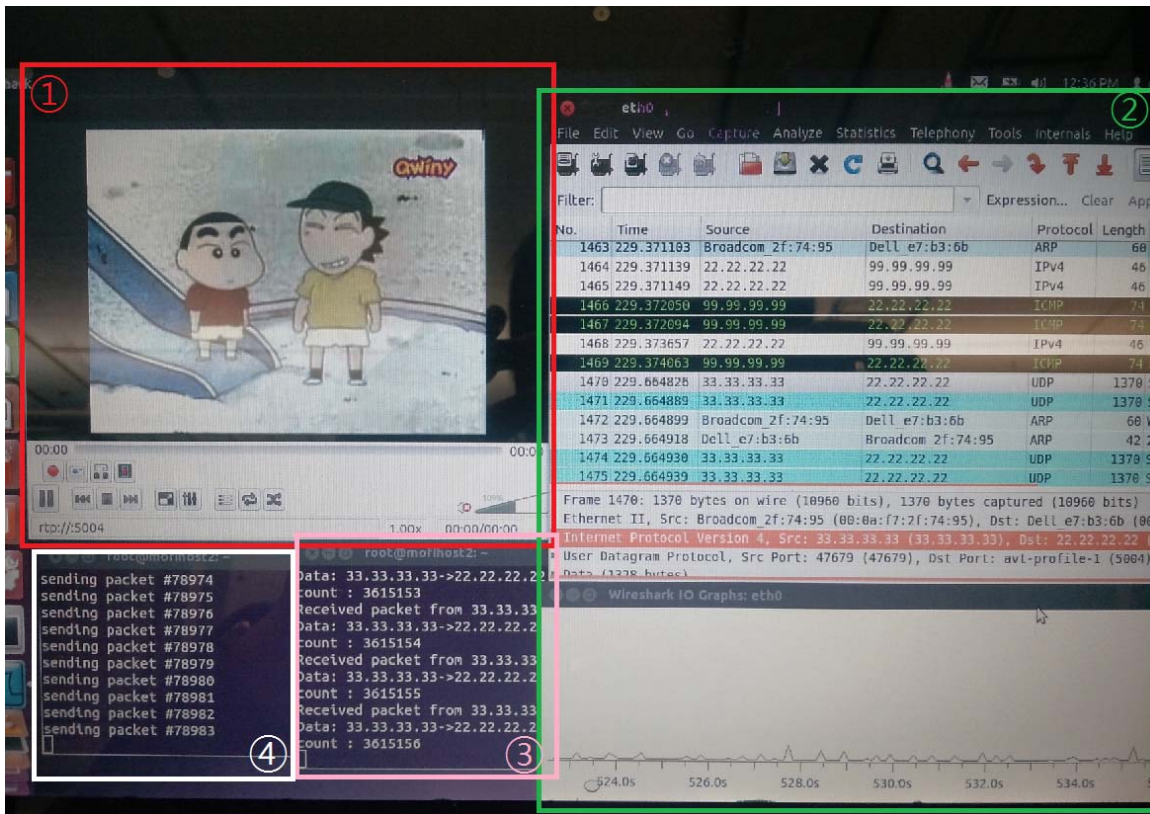


Figure 8. Host 동작

위의 그림은 Host에서 동작을 하여 MOFI 환경에서 프로그램을 실행해본 모습이다. 1번의 경우에는 간단한 동영상 스트리밍을 Host ID를 이용해서 통신하고 있는 모습이다. 2번의 경우에는 Host ID로 통신 되고 있는 패킷을 바탕으로 와이어샤크에서 캡처를 하면서 패킷량을 그래프화 한 모습이다. 3번의 모습은 스트리밍 서비스가 아닌 간단한 패킷 전송 프로그래밍을 통해서 한것으로 각각의 Host ID로 간단하게 전송한 모습이다. 마지막으로 4번의 동작은 Host ID의 Binding Update를 하는 모습이다. 지속적인 업데이트를 통해서 domain의 컨트롤러에게 정보를 업데이트 해주는 모습이다.

## 4. 결론

지금까지 MOFI Testbed를 하기 위한 Openvswitch를 이용하여 만든 Testbed 환경 구축 환경이다. 물론 아직 많은 부분이 MOFI 환경을 다 구현하기 위해서는 부족한 부분들이 존재하지만 그 부분들의 업데이트는 추후 계속해서 진행 될 것이며 앞으로도 간단한 Testbed 구현을 위한 문서를 정리하였다.