

1. Mobile IPv6 테스트를 위한 설정 및 TestBed 구성 매뉴얼

Mobile IPv6 테스트를 위해 총 7대의 PC를 설정하고 각각의 기능에 맞게 설정하기 위한 매뉴얼을 작성한다.

2. 준비 사항

- * 최신 MIPv6 소스 코드 (<http://www.mobile-ipv6.org/>)
- * MIPv6를 위한 커널 소스 (<http://ftp.kernel.org/>)

3. 커널 패치 및 최신 MIPL 설치

Linux에서 Mobile IPv6를 구현한 MIPv6를 설치하기 위해 <http://www.mobile-ipv6.org>에서 최신의 MIPv6 소스 코드를 내려 받는다. MIPv6는 특정 버전의 커널에 최적화 되어 있기 때문에 최신 MIPv6의 버전에 맞게 커널을 수정하고 다시 컴파일 해야 한다.

본 매뉴얼에선 기본적으로 데비안 리눅스(kernel 2.6.12-9-386)를 설치한 상태에서 Mobile IPv6를 위한 커널 컴파일과 MIPv6패치를 하는 과정을 서술한다.

3.1 MIPv6을 위한 커널 패치

- * IPv6 지원 여부 확인
 - 최근의 2.4.x 이상 버전대의 커널을 포함한 대부분의 리눅스 배포판은 IPv6 를 기본적으로 지원하도록 되어있다.
- * 준비한 MIPv6 소스에 맞는 커널을 확인한다.

Current release

- Latest **USER SPACE** part: `mip6-2.0.1.tar.gz`
- Latest **KERNEL** part: `mip6-2.0.1-linux-2.6.15.patch.gz` (only for kernel 2.6.15)

- * 설치한 리눅스 배포판의 커널 보다 상위의 커널 버전을 요구할 경우 해당하는 커널 버전을 내려 받고 파일을 푼다. (커널 소스와 패치는 /usr/src 에 놓는다)

```
# cd /usr/src
# wget http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.15.tar.gz
# tar xvfz linux-2.6.15.tar.gz
# ln -s linux-2.6.15 linux
```

```
# cd linux
```

- * 커널 컴파일 옵션 설정을 위해 menuconfig를 실행시킨다.
 - 데비안 계열의 경우 기본적으로 ncurses가 포함되어 있지 않으므로 apt-get이나 dselect를 이용해 libncurses5-dev 패키지를 찾아 설치해 준다.
 - 옵션 설정을 하기 전에 mrproper를 통해 이전 컴파일 설정을 다 지운다.

```
# apt-get install libncurses5-dev
# make mrproper
# make-kpkg clean
# make menuconfig
```

- * 패치한 커널을 컴파일 한다.
 - 데비안 계열의 경우 커널 패키지를 만들어서 커널을 설치하는 것이 편리하므로 kernel-package를 이용한다.

```
# make-kpkg --revision=2.6.15 --initrd binary-arch
```

- initrd를 설정 하지 않을 경우 kernel Panic이 나는 경우가 생기므로 반드시 포함 시킨다. (initrd 파일이 생성되고 menu.lst 안에 initrd 가 설정된다)
- 새로 만들어진 커널 패키지들은 /usr/src/ 에 생성되고, dpkg를 이용해서 데비안 시스템에 설치할 수 있다.

```
# pwd
/usr/src
# ls
kernel-headers-2.6.15_2.6.15_i386.deb  kernel-image-2.6.15_2.6.15_i386.deb
#dpkg -i kernel-headers-2.6.8_i386.deb
#dpkg -i kernel-image-2.6.8_i386.deb
```

3.2 MIPv6 패치

- * 패치 파일을 적용 시킨다.

```
# pwd
```

```
/usr/src/linux
# zcat mipv6-2.0.1-linux-2.6.15.patch.gz | patch -p1
```

* 에러 없이 패치가 적용 되었다면 커널 컴파일 옵션을 다시 설정 후 커널 컴파일을 한다.

- 커널 패치 파일에 포함된 INSTALL 파일을 참조 하여 mobility와 관련된 옵션을 모두 포함 시킨다.

* 옵션 설정을 모두 마치고 커널 컴파일을 성공적으로 수행하였다면 mipv6 USER SPACE를 설치한다.

- USER SPACE 컴파일시 kernel header를 필요로 하므로 path를 설정해 준다.
- mipv6 2.0부터 노드들의 바인딩 캐쉬나 바인딩 업데이트 리스트를 보기 위한 사용자 툴 대신 virtual terminal을 제공하므로 USER SPACE 설치 시 포함 시킨다.

```
# CPPFLAGS='-isystem /usr/src/linux/include' ./configure --enable-vt
# make && make install
```

* 컴파일과 설치가 성공적으로 수행되었다면 /usr/local/src/mip6-2.0.1/src/에 mip6d (Mobile IPv6 protocol implementation) 데몬인 실행파일이 생성된다.

* 시스템 시작 시 mip6d 데몬이 자동적으로 실행되게 하기 위해 부팅 스크립트에 등록 한다.

- mip6d 데몬 시작 시 해당 노드의 역할에 따라 적절한 파라미터를 주면서 실행 시킨다. (노드가 MN일 경우 mip6d -M, CN일 경우 mip6d -C, HA일 경우 mip6d -H)
- 데비안 계열의 경우 /etc/init.d 디렉토리에 스크립트 파일을 만들고 update-rc.d 명령을 이용해서 현재 시스템의 런레벨에 스크립트 파일 링크를 생성한다. (본 매뉴얼에서는 rc.local란 이름의 스크립트 파일을 생성해서 사용한다.)

```
# cat rc.local
#! /bin/sh
# 데몬 시작시 각 노드의 역할에 맞게 파라미터를 설정한다.
# 현재 노드는 모바일 노드의 역할을 수행한다.
mip6d -M
# update-rc.d rc.local defaults 2
```

4. Test bed 구성

4.1 Routing 엔트리 추가

* Test bed의 구성에 맞게 각 노드들에게 IPv4, IPv6 address를 할당한다. (address는 /etc/network/interfaces 파일을 통해 할당한다.)

- address를 할당한 일부 예로 goliath의 interfaces파일은 다음과 같다.

```
#부팅시 eth1이 자동으로 올라 오도록 설정한다.  
auto eth1  
# interface의 ipv6 address를 static으로 잡아준다.  
iface eth1 inet6 static  
    address 2001:0220:1404::1  
    netmask 64
```

* ip route 명령어를 통해 인터페이스 별로 라우팅 엔트리를 추가 시킨다.

- valkyrie, vulture, medic의 경우 라우팅 기능을 필요로 하지 않으므로 해당 인터페이스에 default gateway 엔트리만 설정한다.

```
ip -6 route add ::/0 via 2001:0220:1404:2::1 dev eth0
```

- goliath은 2001:0220:1404:3::, 2001:0220:1404:2:: 네트워크로 패킷을 라우팅하기 위해 각 네트워크에 대한 next hop(gateway)를 설정해 준다.

```
ip -6 route add 2001:0220:1404:3::/64 via 2001:0220:1404::3 dev eth1  
ip -6 route add 2001:0220:1404:2::/64 via 2001:0220:1404::2 dev eth1
```

- vessel은 2001:0220:1404:3::, 2001:0220:1404:1:: 네트워크로 패킷을 라우팅 하기 위해 각 네트워크에 대한 next hop(gateway)를 설정한다.

```
ip -6 route add 2001:0220:1404:3::/64 via 2001:0220:1404::3 dev eth0  
ip -6 route add 2001:0220:1404:1::/64 via 2001:0220:1404::1 dev eth0
```

- siege-tack는 2001:0220:1404:1::, 2001:0220:1404:2:: 네트워크로 패킷을 라우팅 하기 위해 각 네트워크에 대한 netx hop(gateway)을 설정한다.

```
ip -6 route add 2001:0220:1404:1::/64 via 2001:0220:1404::1 dev eth0  
ip -6 route add 2001:0220:1404:2::/64 via 2001:0220:1404::2 dev eth0
```

* 현재 시스템의 ipv6 라우팅 설정은 route -A inet6 명령문을 통해 확인 할 수 있으며 추가한 각 라우팅 엔트리들은 부팅시 자동으로 추가 될 수 있도록 앞서 만들어 놓은

rc.local 부팅 스크립트에 추가 한다.

* 각 노드들의 ipv4 라우팅 설정도 위와 같이 각각 설정한 후 rc.local 부팅 스크립트에 추가하도록 한다.

4.2 라우터 설정

* 라우터의 인터페이스의 경우 전달 받은 패킷을 다른 네트워크로 전달 할 수 있어야 하므로 forwarding 기능을 enable 시키기 위한 명령문을 rc.local 부팅 스크립트 파일에 추가 시킨다.

```
echo 1 > /proc/sys/net/ipv6/conf/all/forwarding
```

* IPv6를 위한 router advertisement daemon 설정
- apt를 이용해 radvd 데몬을 설치한다.

```
apt-get install radvd
```

- 설치한 radvd 데몬의 설정 파일을 변경하거나 추가 한다. 설정 파일의 각 항목은 radvd.conf man페이지를 참조 하여 Router Advertisement를 주기적으로 전송할 interface와 interval을 설정한다. (radvd.conf man 페이지의 example 참조)

```
interface eth2 {                                # ra메시지를 전송할 interface를 설정
  # route가 주기적으로 RA 메시지를 전송하거나 router solicitation에 대한
  # response를 전송하기 위한 설정
  AdvSendAdvert on;
  # RA 메시지를 전송할 Interval을 설정
  AdvIntervalOpt on;
  MinRtrAdvInterval 3;
  MaxRtrAdvInterval 10;
  # RA 메시지를 전송하는 route가 Home Agent의 기능을 수행한다는 것을 설정
  AdvHomeAgentFlag on;
  # link상의 prefix 정보를 나타냄
  prefix 2001:0220:1404:1::/64 {
    AdvOnLink on; # 설정된 prefix가 link상에서 사용될 수 있다는 것을 설정
    AdvAutonomous on; # stateless address configuration을 할 수 있음을 설정
    AdvRouterAddr on;
  };
};
```

```
};
```

- radvd 설치와 설정 파일이 완료 되었다면 radvd가 시스템 부팅시 자동으로 시작 될 수 있도록 부팅 스크립트에 추가 시킨다.

```
/etc/init.d/radvd start
```

- radvd 메시지들이 주기적으로 보내어지고 있는지 확인하기 위해서 radvdump 명령을 이용할 수 있다.
- * router 역할을 담당하지 않는 노드들은 주기적으로 RA 메시지를 수신해서 address를 자동으로 설정하도록 다음과 같은 명령문을 부팅 스크립트 파일에 추가한다.

```
echo 1 > /proc/sys/net/ipv6/conf/all/autoconf  
echo 1 > /proc/sys/net/ipv6/conf/all/accept_ra  
echo 1 > /proc/sys/net/ipv6/conf/all/accept_redirects
```

4.3 mip6d 데몬 설정

각 노드들은 mobile ipv6를 지원하기 위해 담당할 역할 별로 MN, CN, HA의 기능을 수행할 수 있도록 mip6d.conf파일을 수정한다.

- * 모든 노드에 공통적으로 설정해야 되는 옵션
 - NodeConfig CN | HA | MN; (노드의 역할을 설정, default CN)
 - DebugLevel number; (데몬의 debugging 레벨을 나타냄)
 - DoRouteOptimizationCN boolean; (mobile node와 route optimization을 수행할 것인지를 나타냄, default enable)
 - NonVolatileBindingCache boolean; (Binding cache 정보를 비휘발성 메모리에 저장하기 위한 옵션이나 현재 버전에서는 설정해도 무시됨)
- * HA와 MN에 공통적으로 적용되는 옵션
 - Interface name (각 operation을 수행하는 interface를 설정)
 - {
 - MnIfPreference number; (Interface의 preference를 나타냄)
 - IfType CN | HA | MN; (interface의 역할(MN or CN or HA)을 추가적으로 나타냄)
 - UseMnHaIPsec boolean; (MN-HA MIPv6 signalling을 IPsec을 통해 보호할 것인지를 나타냄, default enabled)
 - KeyMngMobCapability boolean; (

* HA에만 적용되는 옵션

- HaMaxBindingLife number; (MN의 바인딩 업데이트 엔트리의 최대 life time (초 단위), default 262140)

* MN에만 적용되는 옵션

- MnMaxHaBindingLife number; (MN에서 HA로 보내는 바인딩 업데이트 엔트리의 최대 life time(초 단위)
- MnMaxCnBindingLife number; (MN에서 CN으로 보내는 바인딩 업데이트 엔트리의 최대 life time(초 단위)
- DoRouteOptimizationMN boolean; (CN과의 route optimization을 수행할 것인지를 나타냄)
- MnUseAllInterfaces enabled | disabled (mobility를 위해 모든 interface를 사용할 것인지를 나타냄, 일반적으로 이 옵션은 사용하지 않고 Interface 옵션을 통해 명시적으로 사용할 인터페이스를 지정함)
- UseCnBuAck boolean; (CN에게 바인딩 업데이트를 보낼 때 Acknowledge bit를 설정하도록 함)
- MnHomeLink name { (interface가 물리적으로 link에 연결하기 위한 설정 정보들을 저장함)
 - HomeAddress address/length; (ip address와 prefix정보를 설정함)
 - HomeAgentAddress address; (MN의 Home Agent의 address, DHAAD를 사용하려면 unspecified address(::))를 사용함.)
 - MnRoPolicy (route optimization을 위한 정책 설정에 사용됨)

노드 별 역할에 맞게 설정 파일 작성을 완료한 다음 적용을 위해 mip6d 데몬을 다시 시작한다.

5. Mobile IPv6 테스트

상위 단계에서 모든 설정을 완료하고 TestBed를 성공적으로 구성하였다면 Mobile IPv6 테스트를 위해 MN이 handoff를 제대로 수행하는지를 확인한다. MN의 설정은 상위 단계에서 설정된 정보를 그대로 사용하도록 한다.

* RA 메시지 수신

TestBed가 성공적으로 구축되었다면 무선 AP를 통해 2001:0220:1404:3:: 네트워크에 접속한 MN은 router solicitation 메시지 전송 후 router advertisement 메시지를 수신하게 되고 이후 IP address를 자동으로 설정한 것을 확인 할 수 있다.


```
coa 2001:220:1404:2:207:ebff:fe31:7003 flags ---- type bul ack ready RR state ready
lifetime 402 / 420 seq 53547 resend 0 delay 420(after 403s)
```

- handoff시 설정된 ip6tnl1인터페이스가 MN의 default gateway로 설정된 것을 확인한다.

Destination	Next Hop	Flags	Metric	Ref	Use	Iface
::/0	::	U	128	0	0	ip6tnl1

- 추가적으로 CN에서 전송하던 ping 메시지가 끊김없이 전송되는 것을 확인 할 수 있다.