

IPv6 포럼 코리아 기술문서 2001-004

무단복제 금지



IPv6 PC 라우터 및 호스트 설치 및 설정 방법

IPv6 PC Router & Host Configuration (Linux 2.4.X)

이경진 (K. J. Lee) ETRI
이승윤 (S. Y. Lee) ETRI
김용진 (Y. J. Kim) ETRI

목차

1. Linux 운영체제 설치
 2. 네트워크 구조
 3. 네트워크 인터페이스의 IPv6 주소 지정
 4. 터널 인터페이스 설정
 5. IPv6 라우팅 테이블 엔트리 설정
 6. PC 라우터의 작동 및 검사
 7. IPv6 호스트 설치 및 설정
- 참고문헌

IPv6 PC 라우터 및 호스트 설치 및 설정 방법

IPv6 PC Router & Host Configuration (for Linux 2.4.X)

이경진 (K. J. Lee) ETRI
이승윤 (S. Y. Lee) ETRI
김용진 (Y. J. Kim) ETRI

본 문서는 Linux 2.4.X 운영체제를 이용하여, IPv6 Network 구축을 위한 IPv6 PC 라우터와 호스트의 설치와 설정 방법을 설명하고, 터널링(Tunneling)을 통한 다른 IPv6 Network과의 연동 방법을 기술한다.

1. Linux 운영체제 설치

리눅스 커널 버전 2.4.X 는 IPv6를 support하고 있기 때문에 별도의 patch없이 커널 컴파일만으로 IPv6를 지원할 수 있다. 그럼 우선 최신 커널을 다운로드 받아 설치하는 과정을 살펴보자. 다음 서버에 접속하여 최신 버전의 커널을 다운로드 받을 수 있다.

```
ftp://ftp.kr.kernel.org/pub
http://www.kr.kernel.org/pub
ftp://ftp.kernel.org/pub
http://ftp.kernel.org/pub
```

이번 IPv6 network 구성에서 사용한 리눅스 버전은 레드햇(RedHat) 7.1이며 커널 버전은 linux-2.2.7이다. 커널 소스를 받은 후 /usr/src/linux-2.2.7에 압축을 풀고 심볼릭 링크(symbolic link)를 설정한다.

```

/usr/src$ rm -rf linux
/usr/src$ ln -s linux-2.4.5 linux
/usr/include$ rm -rf asm linux scsi
/usr/include$ ln -s /usr/src/linux/include/asm asm
/usr/include$ ln -s /usr/src/linux/include/linux linux
/usr/include$ ln -s /usr/src/linux/include/scsi scsi

```

IPv6를 지원하기 위해서 커널옵션에서 반드시 설정해 주어야 할 것들은 다음과 같다. 이밖에 시스템 특성에 따라 필요한 옵션들을 설정해 주어야 한다. (표 1)은 설정이 필요한 옵션들을 정리하고 있다.

(표 1). IPv6를 지원하기 위해 설정이 필요한 커널 옵션들

커널 옵션		설정유무
Code maturity level options	Prompt for development and/or incomplete code/drivers	Yes
Loadable module support	Enable loadable module support	Yes
	Set version information on all symbols for modules	Yes
	Kernel module loader	Yes
Console drivers	Video mode selection support	Yes
Networking options	Packet Socket	Yes
	Unix domain sockets	Yes
	Kernel/User netlink socket	Yes
	Routing messages	Yes
	TCP/IP networking	Yes
	The IPv6 protocol	Yes
	IPv6: enable EUI-64 token format	Yes removed 2.4.5+
	IPv6: disable provider based address	Yes removed 2.4.5+
File systems	/proc filesystem support	Yes
Kernel hacking	Magic sysrq key	Yes

각 커널 옵션들의 의미는 다음과 같다.

Code maturity level options

Y를 선택하면 개발중인 새로운 기능과 새로운 드라이버를 포함할 것인지에 대해 선택할 수 있도록 하는데, 이 옵션을 켜 두었을 때 시스템 특성에 따라 에러가 발생하기도 한다. IPv6는 `experimental` 버전이므로 이 옵션을 Y로 설정해야 한다.

Loadable module support

리눅스의 모듈기능에 대한 설정.

Enable Loadable module Support : `version 1.2` 이후부터 리눅스 커널은 모듈 기능을 이용한다. 모듈을 이용하면 자주 쓰이지 않는 장치 드라이버나 기능들을 커널 바깥에 모듈로 만들어 두었다가 필요할 때에만 동적으로 메모리에 적재하여 사용하고 작업이 끝나면 메모리에서 다시 제거하므로 메모리를 효율적으로 사용할 수 있고, 커널 크기가 감소한다. 또한, 모듈은 스스로가 컴파일되어 독자적인 기능을 가지므로 모듈로 설정한 기능에 변화가 있더라도 전체 커널에는 손대지 않을 수도 있다. 파일시스템, 장치 드라이버, 바이너리 포맷 등 많은 기능이 모듈을 지원한다. 반드시 [Y]를 선택.

Set version information on all symbols for modules : 다른 버전의 커널에서 만들어진 모듈이나 커널과 함께 배포되지 않는 특별한 모듈을 사용할 수 있도록 하는 기능.

Kernel module Loader : `kerneld` 데몬이 대체된 기능. 모듈로 만들어진 기능을 필요할 때 적재하고 제거하는 작업들을 `modprobe`를 이용해 커널이 자동으로 관리한다. 물론 적절한 옵션들도 덧붙인다.

Networking options

Packet Socket : `tcpdump`처럼 매개 프로토콜 없이 직접 네트워크 장치와 통신하는 어플리케이션에서 사용된다.

Kernel/User netlink Socket : 커널의 어떤 부분들 또는 모듈과 유저 프로세스 사이의 양방향 통신을 허락한다. 유저 프로세스는 `/dev` 디렉토리로 부터 읽거나 쓸 수 있다. `routing message` 기능과 함께 네트워크 관련 정보를 알리기 위해 사용하고 IP: `firewall packet netlink device` 기능과 함께 가능한 공격에 대한 정보를 알리기 위해 방화벽 코드에서 사용합니다. `arpd` 데몬, 네트워크 링크 드라이버를 사용하기 위해서는 [Y]를 선택해야 한다.

Routing messages : 네트워크 관련 라우팅 정보를 `/dev/route`에서 읽을 수 있게 하는 기능이다. `mknod("man mknod")`로 메이저 넘버 36, 마이너 넘버 0인 스페셜 캐릭터 파일로 `/dev/route`를 만들면 이 파일을 읽어서 라우팅 정보에 대해 알아낼 수 있다.

Unix domain Sockets :소켓은 네트워크 연결을 **open**하고 **access**하기 위한 표준 메커니즘이다.. 리눅스 박스가 어떤 네트워크에도 연결되지 않아도 X 윈도우 시스템이나 **syslog** 같은 많은 일반적인 프로그램이 소켓을 사용한다.

TCP/IP networking : 거의 대부분 네트워크에서 사용하는 표준 프로토콜로 인터넷 접속 뿐만 아니라 다른 많은 프로그램에 꼭 필요하므로 반드시 **[Y]**를 선택한다.

The IPv6 protocol (EXPERIMENTAL) : IPv6을 실험적으로 지원한다.

File systems

/proc filesystem Support : 커널과 프로세스를 위한 가상의 파일시스템으로 실제 디스크 공간을 점유하지는 않는다. 프로세스 등 시스템 정보를 제공하는 많은 프로그램이 사용하는 유용한 기능이므로 반드시 **[Y]**를 선택한다. **/proc** 아래 시스템 정보들은 텍스트형태로 관리되므로 **cat** 명령으로도 정보를 볼 수 있다. 예를 들어 **cpu** 정보를 보려한다면 다음 명령을 사용한다: **# cat /proc/cpuinfo**

Console drivers

Video mode selection Support : 커널을 시작할 때 **"vga="**옵션으로 비디오 모드를 선택하는 기능이다. 예를 들어 **"VGA=ASK"**옵션을 사용하면 커널이 시작될 때 비디오모드 메뉴가 나타난다. **"man bootparm"** 명령으로부터 매개변수를 알아볼 수 있다.

Kernel hacking

Magic sysrq key : **[Y]**로 설정할 경우 시스템이 심각한 문제가(**crash**) 있어도 통제할 수 있다. 예를 들면 버퍼 캐시를 디스크로 옮기고, 시스템을 재부팅 하거나 상태 정보를 표시한다. 이 기능은"**<alt>+<SysRQ>**"를 누른 채 **k, r, s** 등 **<command key>**를 눌러 사용할 수 있고 **SysRQ** 키가 없는 키보드는 **PrtSc** 키를 누르면 된다.

위의 옵션들을 설정해준 후 커널 컴파일하고 모듈을 설치한다.

```

/usr/src/linux$ make dep; make clean; make bzImage
/usr/src/linux$ make modules; make modules_install
/usr/src/linux$ cp ./arch/i386/boot/bzImage /boot/vmlinuz-2.4.7

# /etc/lilo.conf 수정

/usr/src/linux$ /sbin/lilo
/usr/src/linux$ reboot

```

이제 IPv6를 설정할 준비가 되었다.

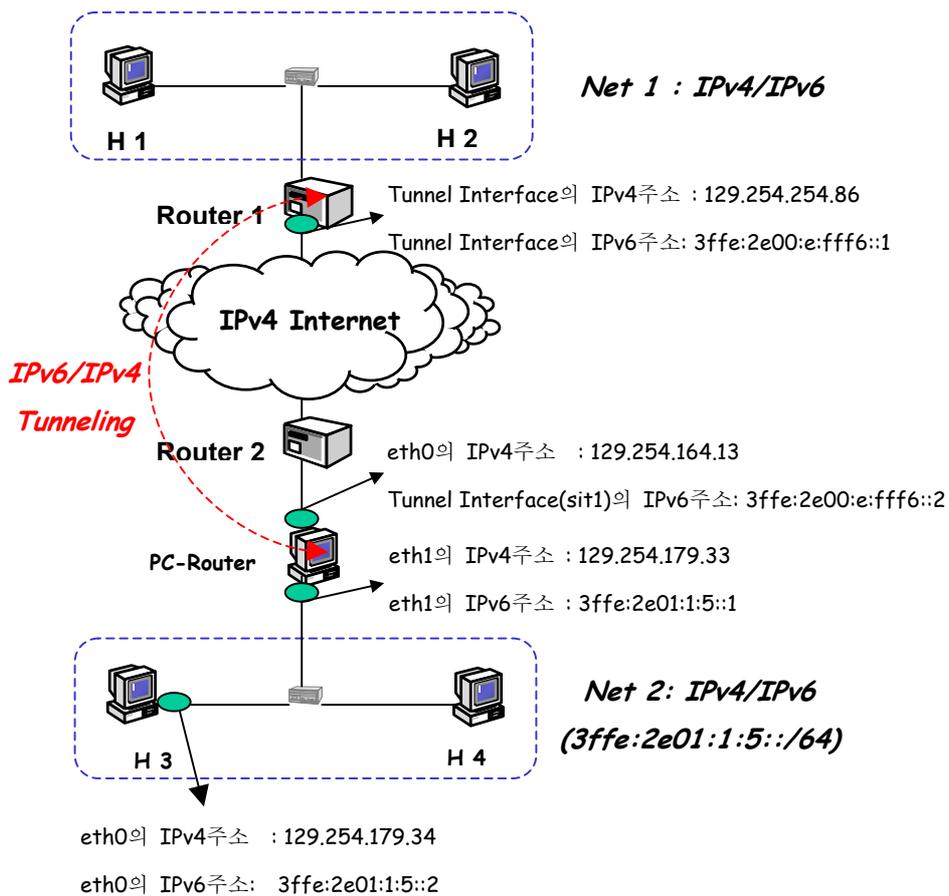
PC 라우터를 구성하므로 두 개 이상의 NIC(Network Interface Card)를 해당 PC에 설치한다. 참고로 본 실험에서는 다음 두개의 NIC를 사용하였다.

eth0 : Realtek RTL 8139(A) PCI Fast Ethernet Adapter
eth1 : 3Com EtherLink XL 10/100 PCI TX (3c509B-TX)

해당하는 NIC 모듈은 Network Device Support 커널 옵션 항목에서 설정해 주어야 한다. (본 문서에서는 NIC 설치에 대한 내용은 기술하지 않는다.)

2. 네트워크 구조

이번 실험에서 구성하고자 하는 네트워크 모델은 (그림 1)과 같다.



(그림 1) 터널링 네트워크 구조

(그림1)과 같이 PC-Router와 Router 1을 터널링 (Tunneling)하고자 한다. PC-Router는 IPv6 Prefix 3ffe:2e01:1:5::/64를 할당 받아서 서브넷 Net2를 구성하고 Router 1과 IPv6 over IPv4 터널링을 한다. Router 1의 TEP(Tunneling End Point) IPv4 주소는 129.254.254.86이고, PC-Router의 TEP IPv4 주소는 129.254.164.13 이다. Router 1의 TEP IPv6 주소는 3ffe:2e00:e:fff6:1이고, PC-Router의 TEP IPv6 주소는 3ffe:2e00:e:fff6:2 이다.

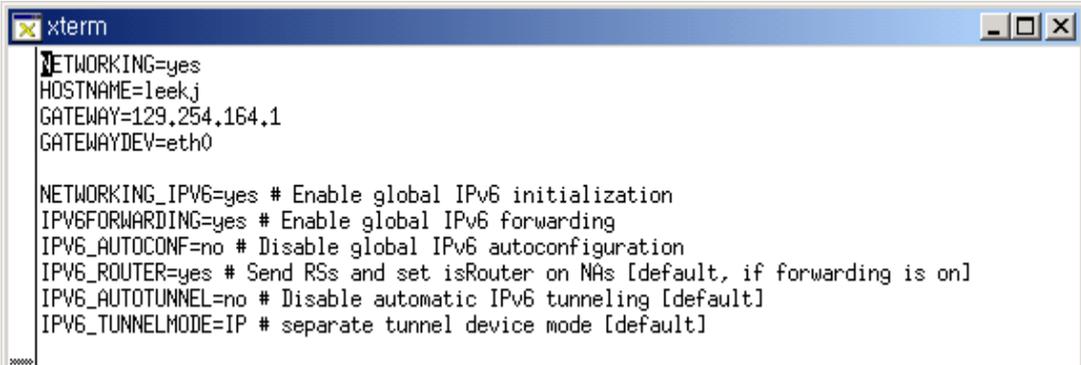
3. 네트워크 인터페이스의 IPv6 주소 지정

Assumption

1. 터널 서버 관리자로부터 할당받은 IPv6 Prefix는 3ffe:2e01:1:5::/64이다.
2. PC-Router의 eth0 IPv4 주소는 Router 1과 터널링되는 IPv4 주소이다.
3. Prefix 3ffe:2e01:1:5::/64의 Subnet은 PC-Router의 eth1과 직접 연결되어 있다.
4. eth1은 Autoconfiguration에 의해 IPv6 주소를 부여 받지만, Prefix 3ffe:2e01:1:5::와 임의의 값(1)을 결합시켜 유일한 IPv6주소 3ffe:2e01:1:5::1을 배정한다. (PC-Router가 관장하는 Link-local Network의 Host가 PC-Router를 default-gateway로 지정할 때 필요하다.)
5. PC-Router의 sit1은 Router 1과 터널링 할 때 이용되는 Tunnel interface이다

Script files 수정

1. 커널 컴파일을 통해서 필요한 기능들을 지원할 수 있도록 하였으므로 이제 스크립트 파일들을 수정하여 IPv6를 enable 시켜야 한다.그럼 먼저 [/etc/sysconfig/network](#) 파일에 다음과 같은 엔트리들을 포함시켜 수정한다.



```

xterm
NETWORKING=yes
HOSTNAME=leekj
GATEWAY=129.254.164.1
GATEWAYDEV=eth0

NETWORKING_IPV6=yes # Enable global IPv6 initialization
IPV6FORWARDING=yes # Enable global IPv6 forwarding
IPV6_AUTOCONF=no # Disable global IPv6 autoconfiguration
IPV6_ROUTER=yes # Send RRs and set isRouter on NAs [default, if forwarding is on]
IPV6_AUTOTUNNEL=no # Disable automatic IPv6 tunneling [default]
IPV6_TUNNELMODE=IP # separate tunnel device mode [default]

```

2. /etc/sysconfig/network-scripts/ 에는 IPv6 지원을 위해 필요한 여러가지 스크립트 파일들이 존재한다. 본 실험에서는 네트워크 인터페이스 카드 두개에 대해서 ifcfg-eth0 와 ifcfg-eth1 스크립트 파일을 각각 수정하여 주었다.

/etc/sysconfig/network-scripts/ifcfg-eth0

```
xterm
DEVICE=eth0
BOOTPROTO=none
BROADCAST=129.254.164.255
IPADDR=129.254.164.13
NETMASK=255.255.255.0
NETWORK=129.254.164.0
ONBOOT=yes

IPv6INIT="yes" # Enable IPv6 initialization of this interface
IPv6TO4INIT="no" # Disable 6to4 initialization of this interface [default]
IPv6_ROUTER=yes # IPv6 autoconfiguration: interface is for routing [default, if forwarding is on]
IPv6_AUTOCONF=no # disable IPv6 autoconfiguration [default, if forwarding is on]
```

/etc/sysconfig/network-scripts/ifcfg-eth1

```
xterm
DEVICE=eth1
BOOTPROTO=none
ONBOOT=yes
IPADDR=129.254.179.33
NETMASK=255.255.255.240
BROADCAST=129.254.179.47

IPv6INIT="yes" # Enable IPv6 initialization of this interface
IPv6ADDR=3ffe:2e01:1:5::1/64 # IPv6 Address
IPv6_ROUTER=yes
```

eth1은 새로운 IPv4/IPv6 서브넷을 구성하는데 IPv4 서브넷은 129.254.179.32~47까지의 범위를 갖도록 설정하였고, IPv6 서브넷은 3ffe:2e01:5::/64 범위를 갖도록 설정하고자 하므로 IPv4 주소는 129.254.179.33으로 설정하고 IPv6주소는 3ffe:2e01:1:5::1로 설정하고 Net2에서 각각의 디폴트 주소가 되도록 하여 라우터 역할을 수행하도록 설정하였다.

4. 터널 인터페이스 설정

Assumption

1. 본 실험에서 Router 1의 IPv4 주소는 129.254.254.86 이다.
2. PC-Router의 터널 인터페이스인 sit1의 물리적 인터페이스 eth0의 IPv4 주소는 129.254.164.13이다.

Notice

리눅스 커널 버전 2.4.x에서는 `ifconfig sit1 up`으로 터널 인터페이스 sit1이 생성되지 않는다. 따라서, sit1을 터널인터페이스로 사용하기 위해서 취할 수 있는 세가지 방법이 있다.

1. automatic-tunneling에 사용되는 sit0에 IPv6 over IPv4 TEP를 할당하면 sit1생성

```
ifconfig sit0 tunnel1 ::129.254.254.86
ifconfig sit1 inet6 add TEP주소(예: 3ffe:2e00:e:fff6::2/127)
route -A inet6 add 3ffe::0/16 dev sit1
route -A inet6 add 2000::0/16 dev sit1
ifconfig sit0 down
```

2. iptunnel 명령어를 이용하여 터널인터페이스 sit1을 생성한다.

```
iptunnel add sit1 mode ipip remote 129.254.254.86 local 129.254.164.13
ifconfig sit1 inet6 add TEP주소(예: 3ffe:2e00:e:fff6::2/127)
route -A inet6 add 3ffe::0/16 dev sit1
route -A inet6 add 2000::0/16 dev sit1
```

3. 모든 것을 스크립트로 해결하고자 한다면, 다음 스크립트를 다운로드 받아서 해당 디렉토리에 복사한 후 이어지는 설명과 같이 수정해 준다.

<http://www.6bone.ne.kr/network-scripts.zip> (/etc/sysconfig/network-scripts/)

Script file 수정

1. IPv6-in-IPv4 터널 인터페이스 sit0,sit1,sit2...인데 sit0는 automatic tunneling을 위해 reserve된 인터페이스 이므로 sitX (X>1)인 인터페이스를 사용한다. 따라서 sit0는 ifcfg-sit0 파일을 다음과 같이 설정하여 사용하지 않았다.

/etc/sysconfig/network-scripts/ifcfg-sit0

```
xterm
DEVICE="sit0"
BOOTPROTO="none"
ONBOOT="no"

IPV6INIT="no" # Initialize IPv6 "yes" or "no"

# This interface script only enable automatic tunneling
# i.e. no other static IPv6 tunnels are configured
# Enable this interface, if you use only automatic tunneling
# otherwise it's not needed
```

2. sit1의 설정 스크립트 파일인 ifcfg-sit1을 다음과 같이 수정하였다.

/etc/sysconfig/network-scripts/ifcfg-sit1

```
xterm
DEVICE="sit1"
BOOTPROTO="none"
ONBOOT="yes"

# Example for a static unnumbered IPv6-in-IPv4 tunnel

IPV6INIT="yes" # Enable IPv6 initialization of this interface
IPV6TUNNELIPv4="129.254.254.86" # IPv4 address of the remote tunnel endpoint [required]
IPV6TUNNELIPv4LOCAL="129.254.164.13" # IPv4 address of the local tunnel endpoint [optional]
IPV6ADDR="3ffe:2e00:e:fff6::2/127" # Local address of a numbered tunnel [optional]
```

5. IPv6 라우팅 테이블 엔트리 설정

이제 필요한 IPv6 루트를 설정해 주어야 한다. 이를 위해서 /etc/sysconfig/static-routes-ipv6라는 스크립트 파일에 필요한 라우팅 테이블 엔트리를 추가한다.

우리가 이미 알고 있는 정보로는 터널 관리자로부터 할당받은 주소블록인

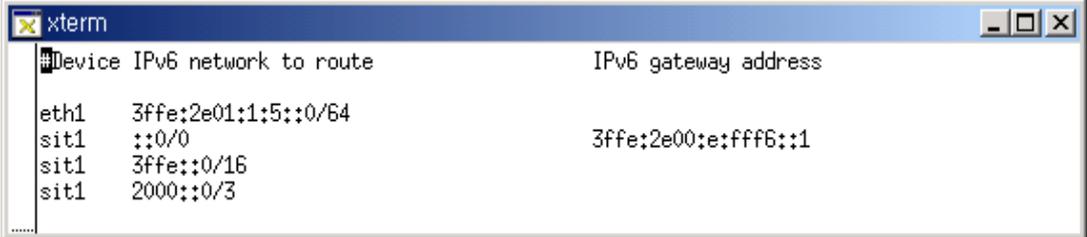
```
3ffe:2e01:1:5::/64
```

일반적으로 알고 있는 실험주소와 공식주소의 주소블록인

```
3ffe::/16
2000::/3
```

이 있으며 이에 대한 루트 정보를 추가해 주어야 한다. 따라서 **eth1**의 IPv6 서브넷인 **3ffe:2e01:1:5::/64** prefix에 매치하는 모든 패킷은 **eth1**을 통하여 전송되도록 하였으며, **longest prefix matching**에 의해 위 실험주소/공식주소 블록에 매치하는 모든 주소가 **sit1**을 통해 라우팅 되도록 엔트리를 추가하였다.

`/etc/sysconfig/static-routes-ipv6`



```

Device IPv6 network to route      IPv6 gateway address
eth1  3ffe:2e01:1:5::0/64
sit1  ::0/0                      3ffe:2e00:e:fff6::1
sit1  3ffe::0/16
sit1  2000::0/3

```

이제 필요한 모든 네트워크 설정을 마쳤다.

6. PC 라우터의 작동 및 검사

이제 모든 수정된 네트워크 구성을 반영해보자.

```
/etc/rc.d/init.d$ network restart
```

명령을 사용하면 재부팅할 필요없이 수정된 네트워크 구성을 실험할 수 있다.

인터페이스 정보 확인

우선 필요한 네트워크 설정이 제대로 이루어 졌는지 확인해보자. 각 인터페이스 정보는 `ifconfig` 명령어로 확인할 수 있다.

1. **eth1**에 IPv6 주소 **3ffe:2e01:1:5::1/64** 가 설정되어있는지 확인한다.
2. **sit1**의 TEP IPv6 주소(**3ffe:2e00:e:fff6::2**)가 설정되어있는지 확인한다.

```

leekj[root]:/etc/sysconfig/network-scripts$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:04:76:6F:3A:E3
          inet addr:129.254.164.13  Bcast:129.254.164.255  Mask:255.255.255.0
          inet6 addr: fe80::204:76ff:fe6f:3ae3/10 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:166815 errors:0 dropped:0 overruns:11 frame:0
          TX packets:6745 errors:0 dropped:0 overruns:0 carrier:1
          collisions:6 txqueuelen:100
          Interrupt:11 Base address:0xc800

eth1      Link encap:Ethernet  HWaddr 00:C0:26:58:8D:06
          inet addr:129.254.179.33  Bcast:129.254.179.47  Mask:255.255.255.240
          inet6 addr: 3ffe:2e01:1:5::1/64 Scope:Global
          inet6 addr: fe80::2c0:26ff:fe58:8d06/10 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:302 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          Interrupt:10 Base address:0x4000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:4089 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4089 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0

sit1      Link encap:IPv6-in-IPv4
          inet6 addr: fe80::81fe:a40d/128 Scope:Link
          inet6 addr: 3ffe:2e00:e:fff6::2/127 Scope:Global
          UP POINTOPOINT RUNNING NOARP  MTU:1480  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0

leekj[root]:/etc/sysconfig/network-scripts$ █

```

라우팅 테이블 확인

라우팅 테이블에서 `static-routes-ipv6` 파일의 루트 엔트리들이 반영되었는지 확인한다.

`route -A inet6` 명령어로 IPv6 라우팅 테이블을 확인할 수 있다.

```

xterm
leekj[root]:~$ route -A inet6
Kernel IPv6 routing table
Destination                                Next Hop                                Flags Metric Ref    Use Iface
::1/128                                    ::                                       U      0      0      0 lo
3ffe:2e00:e:fff6::2/128                   ::                                       U      0      0      0 lo
3ffe:2e00:e:fff6::2/127                   ::                                       UA     256    0      0 sit1
3ffe:2e01:1:5::1/128                       ::                                       U      0      0      0 lo
3ffe:2e01:1:5::/64                         UA     256    0      0 eth1
3ffe::/16                                  ::                                       U      1      0      0 sit1
2000::/3                                    ::                                       U      1      0      0 sit1
fe80::81fe:a40d/128                        U      0      0      0 lo
fe80::204:76ff:fe6f:3ae3/128               U      0      0      0 lo
fe80::2c0:26ff:fe58:8d06/128              U      0      0      0 lo
fe80::/10                                  UA     256    0      0 eth0
fe80::/10                                  UA     256    0      0 eth1
fe80::/10                                  UA     256    0      0 sit1
ff02::1/128                                ff02::1                                UAC    0      1      0 eth0
ff00::/8                                    U      256    0      0 eth0
ff00::/8                                    U      256    0      0 eth1
ff00::/8                                    U      256    0      0 sit1
::/0                                        U      1      0      0 sit1
leekj[root]:~$

```

위의 검사에 이상이 없을 때 IPv6 Networking 기능을 확인해보자.

1. PC-Router 자신의 IPv6주소(3ffe:2e00:e:fff6::2)로 ping6를 실행
2. Router 1의 TEP IPv6 주소(3ffe:2e00:e:fff6::1)로 ping6를 실행
3. Router 1에 터널로 연결된 다른 PC의 TEP IPv6주소(3ffe:2e00:e:ffe::2)를 ping6 실행

```

xterm
leekj[root]:~$ ping6 3ffe:2e00:e:fff6::2
PING 3ffe:2e00:e:fff6::2(3ffe:2e00:e:fff6::2) from ::1 : 56 data bytes
64 bytes from 3ffe:2e00:e:fff6::2: icmp_seq=0 hops=64 time=91 usec
64 bytes from 3ffe:2e00:e:fff6::2: icmp_seq=1 hops=64 time=61 usec
64 bytes from 3ffe:2e00:e:fff6::2: icmp_seq=2 hops=64 time=60 usec

--- 3ffe:2e00:e:fff6::2 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/mdev = 0.060/0.070/0.091/0.017 ms
leekj[root]:~$

```

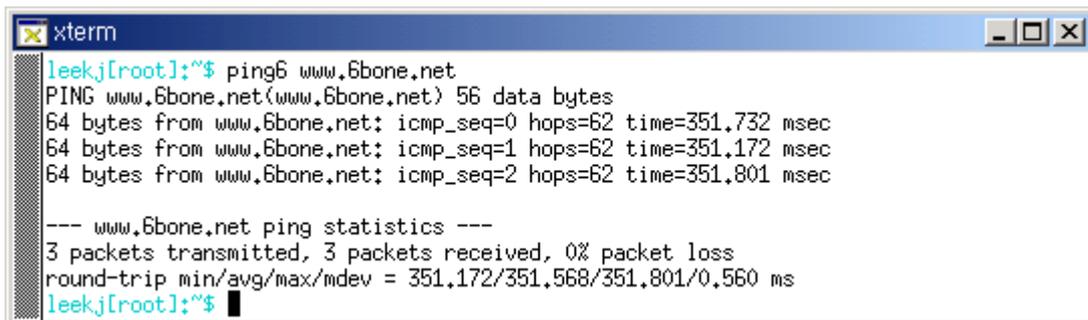
```

xterm
leekj[root]:~$ ping6 3ffe:2e00:e:fff6::1
PING 3ffe:2e00:e:fff6::1(3ffe:2e00:e:fff6::1) from 3ffe:2e00:e:fff6::2 : 56 data bytes
64 bytes from 3ffe:2e00:e:fff6::1: icmp_seq=0 hops=64 time=2.957 msec
64 bytes from 3ffe:2e00:e:fff6::1: icmp_seq=1 hops=64 time=2.528 msec
64 bytes from 3ffe:2e00:e:fff6::1: icmp_seq=2 hops=64 time=2.506 msec

--- 3ffe:2e00:e:fff6::1 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/mdev = 2.506/2.663/2.957/0.215 ms
leekj[root]:~$

```

ping6의 응답이 제대로 돌아오면 IPv6 Networking 기능이 제대로 동작하고 있음을 확인할 수 있다. 그럼 IPv6 DNS 서비스를 받을 수 있는 경우에는 ping6에 IPv6 주소 대신에 도메인 이름(www.6bone.net)을 입력해도 된다.



```

leekj[root]:~$ ping6 www.6bone.net
PING www.6bone.net(www.6bone.net) 56 data bytes
64 bytes from www.6bone.net: icmp_seq=0 hops=62 time=351.732 msec
64 bytes from www.6bone.net: icmp_seq=1 hops=62 time=351.172 msec
64 bytes from www.6bone.net: icmp_seq=2 hops=62 time=351.801 msec

--- www.6bone.net ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/mdev = 351.172/351.568/351.801/0.560 ms
leekj[root]:~$
  
```

7. IPv6 호스트 설치 및 설정

IPv6 호스트를 설치하는 순서

1. 리눅스 설치.
2. 필요한 커널 옵션을 설정하여 커널 컴파일.
3. PC-Router를 설정할 때와 같이 /etc/sysconfig/network와 /etc/sysconfig/network-scripts/ifcfg-eth0 스크립트 파일에 IPv6 Networking에 필요한 구성정보를 추가.
4. 라우팅 테이블에 필요한 루트 엔트리 추가하기 위해 /etc/sysconfig/static-routes-ipv6 스크립트 파일 수정.

IPv6 호스트가 위치할 Network의 IPv6 Prefix와 임의의 값을 가지고 IPv6 주소를 정할 수 있으나, PC 라우터가 Router Advertisement를 통해 Prefix 정보를 공지하므로, Host는 Auto-configuration을 통해 유일한 주소를 할당 받을 수 있다.

Assumption

1. 설치할 호스트는 그림 1에서 Net2의 H3이다.
2. H3의 Network Interface는 eth0라고 가정한다.
3. 그림 1에서 PC-Router가 Prefix 3ffe:2e01:1:5::/64를 Advertise하므로 H3는 EUI-64 방식으로 유일한 IPv6 주소를 배정 받을 수 있다.

3ffe:2e01:1:5:: + Interface ID -> H3의 IPv6 주소

4. Prefix에 임의의 값을 결합하여 IPv6 주소를 배정받을 수 있다.

3ffe:2e01:1:5:: + 임의의 값 -> H3의 IPv6 주소

예) 3ffe:2e01:5::2 /64

Network 설정을 위한 **Script file** 수정

/etc/sysconfig/network

```
NETWORKING=yes
HOSTNAME=leekj3
GATEWAY=129.254.179.33
GATEWAYDEV=eth0

NETWORKING_IPV6=yes
IPV6_AUTOCONF=no
IPV6_AUTOTUNNEL=no
IPV6_TUNNELMODE=IP
```

/etc/sysconfig/network-scripts/ifcfg-eth0

```
DEVICE=eth0
BOOTPROTO=none
ONBOOT=yes
BROADCAST=129.254.179.47
IPADDR=129.254.179.34
NETMASK=255.255.255.240

IPV6INIT=yes
IPV6ADDR=3ffe:2e01:1:5::2/64
```

라우팅 테이블 엔트리 추가를 위한 **Script file** 수정

/etc/sysconfig/static-routes-ipv6

#Device	IPv6 network to route	IPv6 gateway address
eth0	::0/0	3ffe:2e01:1:5::1
eth0	3ffe:2e01:5::0/64	

네트워크 prefix 3ffe:2e01:5::/64 에 매치하는 모든 패킷은 eth0를 통하여 나가도록 하였으며 디폴트 주소는 PC-Router의 IPv6 주소인 3ffe:2e01:1:5::1로 eth0를 통해 나가도록

설정하였다.

설정 확인

인터페이스 정보 확인 : ifconfig

```
eth0  Link encap:Ethernet  HWaddr 08:00:46:0D:B0:48
      inet addr:129.254.179.34 Bcast:129.254.179.47 Mask:255.255.255.240
      inet6 addr: 2001:230:201:1:a00:46ff:fe0d:b048/64 Scope:Global
      inet6 addr: fe80::a00:46ff:fe0d:b048/10 Scope:Link
      inet6 addr: 3ffe:2e01:1:5::2/64 Scope:Global
      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
      RX packets:457 errors:0 dropped:0 overruns:0 frame:0
      TX packets:9 errors:0 dropped:0 overruns:5 carrier:0
      collisions:0 txqueuelen:100
      Interrupt:9

lo    Link encap:Local Loopback
      inet addr:127.0.0.1 Mask:255.0.0.0
      inet6 addr: ::1/128 Scope:Host
      UP LOOPBACK RUNNING  MTU:16436  Metric:1
      RX packets:7 errors:0 dropped:0 overruns:0 frame:0
      TX packets:7 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:0 |
```

IPv6 routing table 확인

```
Kernel IPv6 routing table
Destination      Next Hop          Flags Metric Ref    Use Iface
::1/128          ::                U        0      0      0 lo
3ffe:2e01:1:5::2/128
3ffe:2e01:1:5::/64
fe80::a00:46ff:fe0d:b048/128
fe80::/10        ::                U        0      0      0 lo
ff00::/8         ::                UA       256    0      0 eth0
::/0
::/0             3ffe:2e01:1:5::1 UG       1      0      0 eth0
```

참고문헌

- [1] http://kldp.org/KoreanDoc/html/2.4Kernel_Compile-KLDP/2.4Kernel_Compile-KLD.html#toc22
- [2] <http://www.bieringer.de/linux/IPv6/IPv6-HOWTO/>
- [3] 신명기외 5명, “IPv6 시험망 구축 방법” IPv6 포럼 코리아 기술문서 2001-001