

Praktikum RouterLab SS 2008 Work Sheet 7: Internet Protocol Version 6 (IPv6)

This work sheet introduces Internet Protocol Version 6 (IPv6). After getting familiar with IPv6 addresses, we will study IPv6 Stateless Address Autoconfiguration. Finally, we will extend the home network of the previous two work sheets with IPv6 functionality: Using the 6to4 tunneling technique, we establish “IPv6 connectivity” over an IPv4-only network.

Table 1: Assignment of devices to groups

Gruppe	Ham-Cloud	Muc-Cloud
Router	ham-rj1, ham-rc1	muc-rj1, muc-rc1
Switches	ham-sc1	muc-sc1
Core Network IP range	10.1.0.0/16	10.2.0.0/16
Dialin IP range	105.1.0.0/16	105.2.0.0/16
DataCenter IP range	106.1.0.0/16	106.2.0.0/16
Dialin Wallplug	15A8:01:07	15A8:01:08
IPv6 range	2001:11::/32	2001:12::/32
Loadgens	loadgen104,loadgen106	loadgen102, loadgen107

Question 1: (15 Points) *IPv6 Addresses*

- Name at least four differences between IPv4 and IPv6 (see e.g., RFC 2460)!
- Get familiar with IPv6 addresses by reading RFC 4291 or by consulting other IPv6 literature! Answer the following questions:
 - What are *link-local* addresses and what IP address range do they use?
 - What is the standard IPv6 loopback address?
 - Which address blocks are globally routable and currently assigned by the IANA?
 - What is the complete/long text representation of 2001:0::0:1:2?
- Log on to loadgen104 (loadgen102) and run `ifconfig`. Which *link-local* IPv6 address has been assigned to the interface `eth0`? Explain how this IP address has been derived from the MAC address. For more information read Section “Links or Nodes with IEEE 802 48-bit MACs” of RFC 4291 or look for other sources in the web.

Question 2: (10 Points) *Preparations*

- Restore the setup of Question 4 of Work Sheet 5¹. You don’t need to run Radius but you must run your *own* PPPoE server on your *loadgen*. Also check whether there is a default route from the WRT to the PPP-AC. Before continuing, make sure that the WRT obtains an IP address via PPPoE and that your client (laptop) can download the website `http://106.1.42.42` (`http://106.2.42.42`) via IPv4.
- Activate IPv6 support for the WRT. Log on to the WRT via `ssh` and make sure that the following modules are installed:

¹Don’t forget to enable IP forwarding on Linux machines

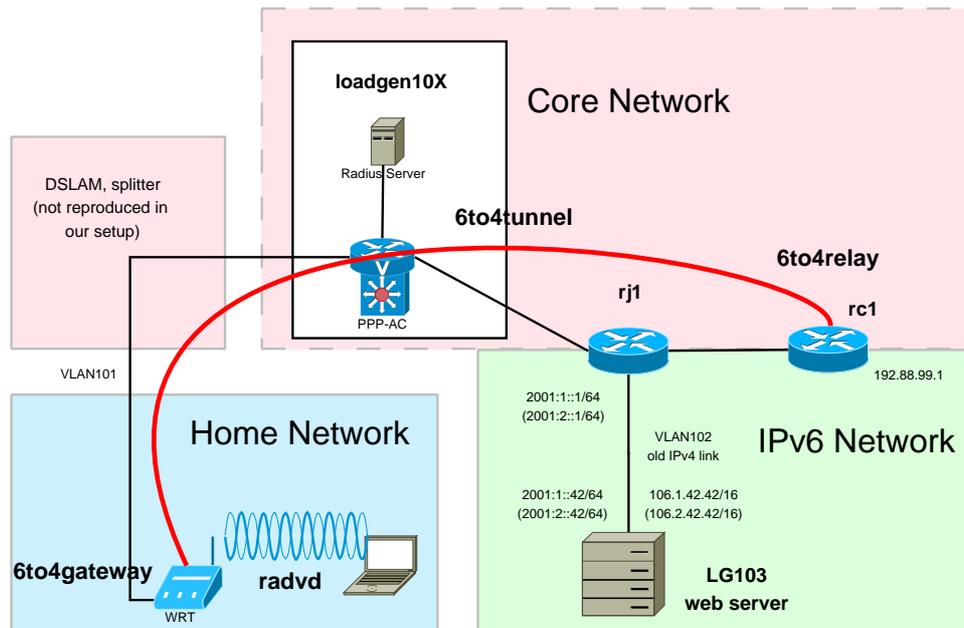


Figure 1: Topology for this assignment

```
ipkg install kmod-ipv6 ip radvd
```

Then load the IPv6 kernel model:

```
insmod ipv6
```

Which IPv6 loopback address has been assigned?

- (c) Activate IPv6 support for your client. On Linux or UNIX systems, this should work out of the box (you should see IPv6 addresses in `ifconfig`). There is plenty of information in the web on how to enable IPv6 for different operating systems! For Windows XP IPv6 you may need the following command:

```
netsh interface ipv6 install
```

Question 3: (50 Points) *Migrating your home network to IPv6*

The web server has now been upgraded and can be additionally reached under the IPv6 address `2001:1::1` (Muc-Cloud: `2001:2::1`). Unfortunately, our provider does not assign an IPv6 address to the WRT.

In the following, we will use *6to4* to talk to our web server using IPv6 (see Figure 1). The WRT acts as a *6to4 gateway* which encapsulates IPv6 packets that it receives from its clients. These packets are then forwarded to a *6to4 Relay* inside the provider network. In our simplistic setup the *6to4 Relay* will be configured on `rc1`, while the *6to4 gateway* will be on the WRT. Packets do not have to know the IPv4 address of a *6to4 Relay* as there exists the well-defined fixed Anycast address `192.88.99.1` that is used for tunnel endpoints, i.e. the *6to4 Relays*. The *6to4 Relay* decapsulates and sends the IPv6 packet to the web server. Draw a topology map where you incrementally add IP addresses that you will assign throughout this question.

- (a) Search the web and briefly explain each of the following concepts in 2 or 3 sentences:
- Tunneling IPv6 over IPv4
 - Automatic Tunneling, *6to4*
 - Configured Tunneling, tunnel brokers, *6in4*
- (b) Generally, it is possible to manually configure static tunnels. What would be the disadvantage in our example? Note that the WRT obtains a dynamic IP address.
- (c) Explain how *6to4* IPv6 addresses are derived from IPv4 addresses (see RFC 3056 Section 2 or <http://linux.die.net/man/8/radvd>)!
- (d) We will now start with the configuration of the *6to4 gateway*. What are appropriate *6to4* addresses for the tunnel endpoint (TUNNEL-*6to4*) and LAN interface (LAN-*6to4*) of the WRT? To find out you may write the following shell script², make it executable (`chmod`) and finally execute it:

²To edit files on the WRT you can either use `vi` if you access the WRT via console/ssh or use the editor of the web interface (System → File Editor). In the second case, `Page 2 of 4` touch "file name" on the console.

```

WANIP=$(ip -4 addr show dev pppX | awk '/inet/ {print $2}' | cut -d/ -f1)
echo "IPv4 address: $WANIP"
V6PREFIX=$(printf '2002:%02x%02x:%02x%02x' $(echo $WANIP | tr . ' '))
TUNNEL6to4=$V6PREFIX:0::1/64
LAN6to4=$V6PREFIX:1::1/64
echo "Tunnel6to4: $TUNNEL6to4, LAN6to4: $LAN6to4"

```

Try to understand the script from above (check `manpages` if necessary). Explain in a few sentences what the script does and add the calculated IPv6 addresses to your topology map. For more information we refer to RFC 3056 (Section 2) or other sources in the web.

- (e) Configure the 6to4 tunnel, IP addresses and routing on the WRT, i.e. the 6to4 gateway! To do this manually, you need the following commands:

```

ip tunnel add tun6to4 mode sit ttl 255 remote any local <IPv4 address of pppX>
ip link set tun6to4 mtu 1280
ip link set tun6to4 up
ip addr add <TUNNEL6to4 address> dev tun6to4
ip addr add <LAN6to4 address > dev br0
ip -6 route add 2001::/3 via ::192.88.99.1 dev tun6to4

```

Explain what each *individual* command does! Check with `ip -6 tunnel show` and `ifconfig` whether the tunnel has been created and if the IPv6 addresses have been assigned!

- (f) *Optional:* Note that after closing and restarting a PPPoE link, you need to redo the configuration steps above. If you want to obtain 5 extra points, you can automatize this task.

Whenever the `ppp` interface comes up, the `ppp` daemon calls the script `/etc/ppp/ip-up` script which again calls any executable script that is placed in `/etc/hotplug.d/iface/''`. Add the required commands to `etc/hotplug.d/iface/*` (see previous two scripts). Here is a skeleton:

```

source /etc/functions.sh
if [ "$ACTION" = "ifup" -a "$INTERFACE" = "wan" ] ; then
    ... add commands here ...
fi
if [ "$ACTION" = "ifdown" -a "$INTERFACE" = "wan" ]; then
    ... remove tunnel etc ...
    [ -f /var/run/radvd.pid ] && kill -HUP $(cat /var/run/radvd.pid)
fi

```

- (g) Enable IPv4 connectivity between `rj1` and `rc1` (use direct link). Assign IPv4 addresses (from the pool `10.X.0.0/16`). Use `ping` to make sure that the setup works.
- (h) Add IPv6 connectivity between `rj1` and `rc1`. Assign IPv6 addresses from the address pool `2001:11::/96` (`2001:12::/96`). Find out how to assign IPv6 addresses for Cisco and Juniper routers (Hint: On Cisco IPv6 needs to be enabled with `ipv6 unicast-routing`, on Juniper just specify an address inside family `inet6`). Use `ping/ping6` to make sure that the setup works.
- (i) Enable IPv6 connectivity between `rj1` and the webserver. `loadgen103` has the address `2001:X::42` and expects `2001:X::1` as its default gateway. Configure `2001:X::1/64` on `rj1`. Don't modify the setup of `loadgen103`. Check with `ping/ping6` if the web server is reachable from `rj1`!
- (j) Now configure `rc1` as a *6to4* relay. It must decapsulate packets received on the "Anycast" address `192.88.99.1` and forward them to the correct host in the IPv6 network. In the reverse direction, it must accept IPv6 packets with address `2002::/16`, encapsulate them and send them to the appropriate IPv4 address. The following steps are required:

Assign the IPv4 address `192.88.99.1/24` to a loopback interface of `rc1` (this IP is a well-known Anycast address to detect *6to4 relays*). What is the corresponding 6to4 IPv6 address?

Configure the tunnel on `rc1` and explain the following commands:

```

interface Loopback0
    ip address <IPv4 loopback address> 255.255.255.255
    ipv6 address <6to4 loopback address/128>
    ipv6 mtu 1480
!
interface Tunnel0
    no ip address
    no ip redirects
    ipv6 unnumbered Loopback0

```

```

tunnel source Loopback0
tunnel mode ipv6ip 6to4
tunnel path-mtu-discovery
!
ipv6 route 2002::/16 Tunnel0

```

- (k) Enable OSPF on rc1 and extend the OSPF setup of rj1. Your *loadgen* (PPP-AC) and rj1 must learn a route to 192.88.99.1. Check with ping if you can reach 192.88.99.1 from your *loadgen*.
- (l) Enable IPv6 reachability between rc1 and the web server. Specify the interface IP (not loopback) of rc1 as source for the ping and check if pinging the address 2001:X::42 from rc1 works. We recommend to add two static routes for this purpose:

```

rc1: ipv6 route 0::/0 <neighbor interface IP on rj1>
rj1: set routing-options rib inet6.0 static route 0::/0 next-hop <neighbor int. IP on rc1>

```
- (m) Check whether you can ping6 from your WRT to 2001:X::42. If ping6 does not work, try to debug your setup systematically, e.g., by running tcpdumps on intermediate hops/routers!

Question 4: (25 Points) *IPv6 Stateless Address Autoconfiguration*

- (a) Have a look at RFC 2462. In particular, read Section 4 “Protocol Overview”!
- (b) Configure your *Linksys WRT-54GL* home router such that it automatically assigns IPv6 addresses to your client (e.g., your laptop) in your home network! The goal is to advertise 6to4 prefixes (see RFC 3056, Section 2).

For this purpose you need the router advertisement daemon *radvd* on your WRT. The following config (*/etc/radvd.conf*) tells *radvd* to announce a 6to4 prefix to an interface. Figure out whether *eth1*, *eth0*, *vlan0*, *br0* etc. is the appropriate interface.³

```

interface <interface name> {
  MinRtrAdvInterval 3;
  MaxRtrAdvInterval 4;
  AdvLinkMTU 1280;
  AdvSendAdvert on;
  prefix 0:0:0:1::/64 {
    AdvOnLink on;
    AdvAutonomous on;
    AdvValidLifetime 300;
    AdvPreferredLifetime 120;
    Base6to4Interface ppp0;
  };
};

```

Finally start the daemon with:

```
/etc/init.d/S51radvd start
```

Run *tcpdump* on the interface of your client (laptop). What do you see? Which packets belong to IPv6 autoconfiguration and which ones are used to assign IPv4 addresses? What is the difference? What IPv6 address has been assigned to your client via IPv6 Stateless Autoconfiguration?

- (c) Download the web site from 2001:1::42 (Muc-Cloud: 2001:2::42)! In case, you want to use *wget* for the download, specify the IPv6 address in square brackets, e.g.: *wget http://[2001:X::42]*.
- (d) Start *tcpdump* on the interface of your *loadgen* that connects with rj1. Use the *-s0 -w <filename>* option to dump a file. Then download the web site from 2001:1::42 (2001:2::42). Stop the *tcpdump* and copy the dumped file to a Linux machine (e.g., on any Snakepit machine in Asterix). There use *wireshark* and try to find the packet with the HTTP Request or Reply. Explain which protocol headers you see in each packets and in which sequence they appear!

For submission details please check the FAQ:

http://www.net.t-labs.tu-berlin.de/teaching/ss08/RL_labcourse/

Due Date: Friday, June 27th, 2008, 08:00 am

³To check for available options look at the *radvd.conf* manpage: <http://linux.die.net/man/8/radvd>