

# Helsinki University of Technology

Department of Electrical and Communications Engineering  
Networking Laboratory

**S-38.3133 – Networking Technology, laboratory course**

**Spring 2007**

**Work number 30: IPv6**

Preliminary exercises, laboratory assignment, questions for final report.

29.7.2003

Niko Suominen

[niko@netlab.hut.fi](mailto:niko@netlab.hut.fi)

Updated 29.7.2004

Oskari Simola

[simola@netlab.hut.fi](mailto:simola@netlab.hut.fi)

Updated 1.8.2005

Eero Solarmo

[eero@netlab.hut.fi](mailto:eero@netlab.hut.fi)

Updated 3.1.2007

Anni Matinlauri

[amatinla@cc.hut.fi](mailto:amatinla@cc.hut.fi)



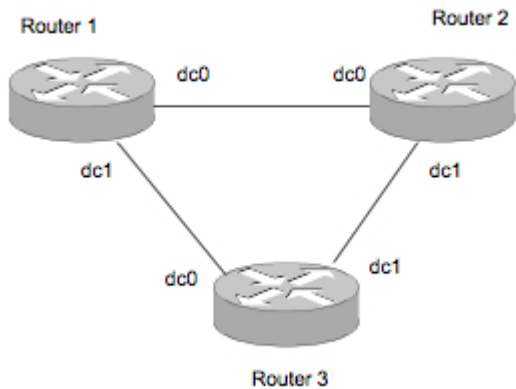
# IPv6 Laboratory Work

## Preliminary Exercises

*Answer the following questions shortly but clearly. Remember to mention your sources. You can answer in Finnish or in English. Native Finnish speakers should answer in Finnish. Make sure you understand the basics of IPv6 addressing before entering the lab. It could also be a (really) good idea to examine the laboratory assignment beforehand. There is only 3 hours reserved work time on your lab turn.*

1. It has been said that the new IPv6 protocol corrects many problems that appeared with IPv4. What are the main advantages of the IPv6 protocol? If the IPv6 really is better than the IPv4, why don't we use it yet in the Internet?
2. In the IPv4 architecture there are three different kinds of addresses: unicast, multicast and broadcast ones. The IPv6 introduces a new type called anycast. What is the purpose of anycast addressing and what are the restrictions of using anycast addresses?
3. a) Explain the following address terms:
  - 1) a global address
  - 2) a site local address
  - 3) a link local addressb) Is it possible to build routing using only link local addresses? (Assume that at least the originating and the destination nodes have a site local or a global address configured.)
4. Explain the differences between stateless and stateful autoconfiguration?
5. The IPv6 introduces also a new version of ICMP protocol. What are new properties of ICMPv6 compared to ICMPv4 protocol?
6. The address notations: RFC2373 (<http://www.ietf.org/rfc/rfc2373.txt>) describes the IPv6 addressing architecture. Study it and answer the following questions.
  - a) What is the IPv6 loopback address? (Just give the address.)
  - b) What address scope do the following addresses belong to?
    - 1) fec2::2
    - 2) ff15:2:5:fece::
    - 3) 3ffe:1:4:3:efa::
  - c) Are the following address notations legal?
    - 1) 1::
    - 2) fec0:0:0:4::2:4
    - 3) ff::ff:345::3
    - 4) ::
    - 5) 3ffe::130.233.220.31
  - d) What does the *IPv4 compatible IPv6 address* mean?

7. a) How do link local addresses start?  
 b) How do site local addresses start?  
 c) You have a tiny network of three routers and each has two network interfaces dc0 and dc1. The routers form a triangle topology. Allocate site local addresses to your network.



<b>Router 1</b>	Address
dc0	?
dc1	?
<b>Router 2</b>	Address
dc0	?
dc1	?
<b>Router 3</b>	Address
dc0	?
dc1	?

8. The header mechanism in the IPv6 protocol is different from the IPv4. There are main and extension headers. What extension header types have been specified and what are the advantages of this mechanism?

9. A router sends an ICMPv6 router advertisement to the network. The authentication header is present. The IPv6 packet is also tunnelled through a link in an IPv4 packet. Draw a picture of the IPv4 packet, which is sent. Include all IP and ICMP headers to the picture. Estimate how many percent of packet data is used to the headers (ICMP headers are not counted)? Make the necessary assumptions. Give an example situation when router advertisements could be sent over many hops.

## **IPv6 Laboratory Work – Laboratory Assignment**

### **Introduction**

In this laboratory work you will familiarize yourself with the IPv6 protocol and with an IPv6 routing protocol, RIPng. Your task is to make the network to work with the IPv6 by configuring routers and hosts. You should be familiar with the basic UNIX commands and with the basic routing IPv4 protocols. It is really recommended that you have done the IPv4 routing assignment before you begin this work. You should also understand the basic differences between the IPv4 and the IPv6 protocols and the addressing scheme. Making the preliminary exercises well will help you in your laboratory work.

After you have completed all the tasks in the laboratory you have to make a final report. You only have to answer questions, which are located in different places of this document.

Your grade depends on the preliminary exercises, on the final report and also on your laboratory work. The IPv6 is still quite experimental, so don't feel down, if everything does not work as you expected. The main thing is to get a view to the IPv6 world.

### **Work Environment**

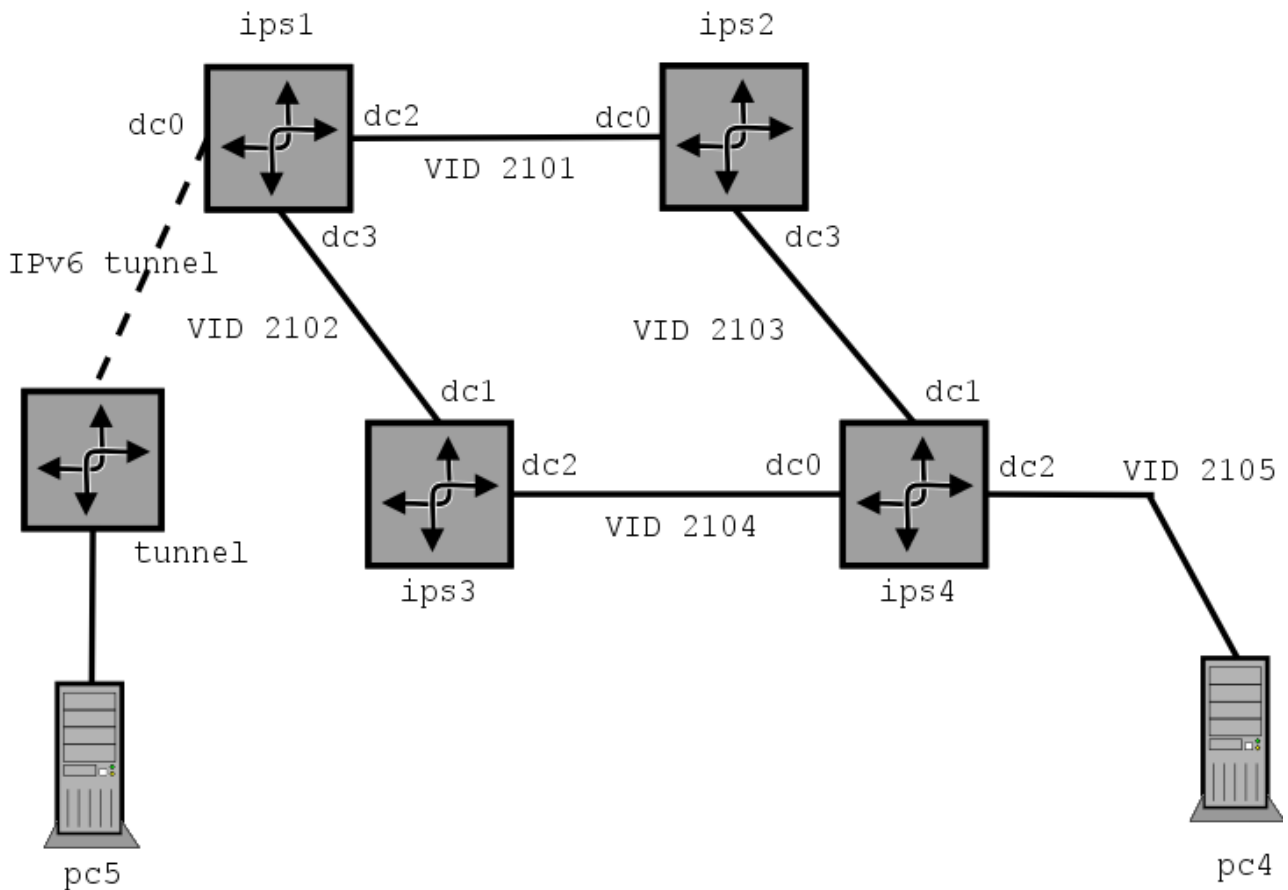
There is a network with 4 routers. The network should have been configured to work with IPv4 addresses, when you enter the laboratory. Routers' domain names should also exist in the lab's DNS server. Figure 1 describes the most important parts of the network. Solid lines are physical links. Lines marked with dots are virtual connections, which you are going to create.

The routers are ordinary PCs running the *FreeBSD* operating system. To connect to an individual router you can use *ssh* or *telnet*. Before the network has been configured with IPv6 addresses you must use IPv4 addresses. You can also use the domain names to connect to the routers by SSH. It's also possible to use console port to access the routers. Every router has been connected to a terminal server and has DNS name ex. ips1.ts.lab. (To connect to ips4 write **telnet ips4.ts.lab**)

Ask from the assistant the user IDs and the passwords for the routers. When doing changes to configurations, be very **careful**. Wrong configurations are sometimes very slow to solve. Notice also that the *tunnel* is not physically connected to the same network as the other routers. You have to make a virtual tunnel over the IPv4 network to make IPv6 connections to *tunnel* and then to *pc5* possible. If you are insecure about the physical connections between nodes, the assistant will help you.

There are four PCs connected to this test network. Some of them have Windows 2k as an operating system. One machine, *ohmi*, has Debian Linux distribution. One goal of the work is to make routing work so that you can connect from any PC to any other PC in the test network using IPv6 protocol.

The IPv6 protocol has been enabled in every router so you don't have to do it again. The Windows machines have also a *Microsoft IPv6 preview* package installed so PCs should also work with the IPv6 protocol. The *ohmi* has a program called Ethereal, which you can use to capture network traffic from any switch port. The program starts with the command *ethereal*. Passwords and other help about using *ohmi* you get from the assistant. Make sure that the dlink switch has right mirroring setup before capturing traffic. Port mirroring setup can be changed from <http://admin.noc.lab/dlink/capture.php>



**Figure 1. The network topology**

## Using FreeBSD

There is some additional FreeBSD related material that you get with this laboratory assignment. The material should help you in your work. Scan them through for more help and take the material with you to the laboratory. This is a short summary of using FreeBSD.

In every FreeBSD router there is a file called `/etc/rc.conf`. This is the main configuration file. Do not change any line in this file unless you know what you are doing.

To get IPv6 routing to work, `rc.conf` file should have at least the following lines:

```

ipv6_enable="YES"
ipv6_gateway_enable="YES"
rtadvd_enable="YES"

```

You don't have to set these lines yourself, because they should already be there. But make sure you understand what they mean. Check that there are no `#`-signs before the configuration lines in the file. They are used to mark comments.

**Question group 1. Explain the meaning of the previous configuration lines? What are the differences between host and router configurations?**

To set an IPv6 address to an interface you can use the `ifconfig` program. Look at the man pages or at other material to get familiar with its syntax. Configurations are active as soon as you have

executed the *ifconfig* program. Remember that, if you set IP addresses with *ifconfig*, they disappear when you reboot the machine.

Another way to configure IP addresses is to edit */etc/rc.conf* file. There are model lines that show the syntax of configuration lines. After editing *rc.conf* you should reboot the machine to get the changes to work. Another way is to connect to the router via console and to type “init 1” to get to the single user mode and then to exit to get back to the multi-user mode. While setting IP addresses make sure you set them to the right interfaces.

To see a router’s routing tables write *netstat -r*. To ping with IPv6 use *ping6* program and to trace route use *traceroute6*. The corresponding commands in Windows 2000 are *ping6* and *tracert6*.

In this work we use **RIPng** (RIP new generation) routing protocol. There is a program called **route6d**, which is an implementation of the RIPng. You have to start this program in every router when your address configurations have been made. The program is very easy to use. Check the man pages or other manuals to get more information. (Tip. You are using site local addresses.)

In Windows machines there is also a command *ipv6*. With the parameter *if* it shows the state of IPv6 interfaces. To get help give the command without parameters.

IPv6 support in Windows 2000 is not a default property. Windows XP is the first Windows, which has IPv6 support as a basic property. IPv6 preview package to Windows 2000 can be found at Microsoft’s home page. Everything you need in the lab has been already installed.

### ***Additional Material***

There are Finnish and English guides how to configure IPv6 in FreeBSD. Material is not identical. Both the guides should be enough for this laboratory work. It’s recommended that you use the Finnish version, if you can.

## **Mission Possible – Exercises in the Laboratory**

### **1. Decide how to allocate IPv6 addresses**

Your first mission is to decide how to use your site local IPv6 address space. The address space should be large enough. Write down the IP addresses you decide to use with certain interfaces.

With *ifconfig* you get a lot of information about available interfaces. Interfaces can be named like *dc0*, *dc1*, *x10*, *x11* etc. After passing the IPv4 laboratory assignment you should be familiar how to use *ifconfig*.

#### **Question group 2. What is DAD?**

Show your IPv6 configuration (on paper) to the assistant before proceeding. In this phase you can forget the IPv6 tunnel. Do not configure the link between *tunnel* and *ips1* yet. Configure only the physical links.

#### **Question group 3. Express your final IPv6 address allocation. Why did you choose addresses like you did?**

### **2. Configure IPv6 addresses to the interfaces**

Use SSH connections to log into the routers. You can use *ifconfig* to configure IP addresses or you can edit *rc.conf* file. Be careful.

Test with *ping6* that the links work. Routing should not work at this phase yet, because there is not any IPv6 routing protocol running.

### **3. Configure the IPv6 in IPv4 static tunnel**

Now you should configure a tunnel between *tunnel* and *ips1*. This is not difficult, but you should find out how to do it. Remember that the tunnel is a point-to-point link and the both endpoints must have an IPv6 address in a same subnet. In other words, you are going to create virtual interfaces to the both endpoints.

#### **Question group 4. Tell how you configured the static tunnel. How does the tunnel really work? Are there any other alternatives to transfer IPv6 traffic in an IPv4 network? What are the disadvantages of tunnels?**

Now you can test again with *ping6* that your virtual link works.

### **4. Start the routing**

Before doing anything you need to start the Wireshark analyser and configure it to capture from relevant places. To do this, follow these instructions:

1. Go to <http://switchcontrol.noc.lab/> and select correct ports from where to capture traffic.
2. Connect to the capture machine with `ssh -X capture@capture.noc.lab`. The password is `capture`. Click `capture->interfaces->(eth1)prepare`. You also need to input a capture filter so that you are only capturing the interfaces. This can be done by writing `ether[14:2] = vlanid1` or `ether[14:2] = vlanid2`. `Vlanid`'s can be found from the `switchcontrol`-pages and some of them can also be seen in the picture of the network topology.



To test that you have correct settings, try ICMP echo requests from *ips1* to *tunnel*.

Now it's time to start the **route6d** program. Start it in every router. Use the right switches. Make it to advertise also static routes. Capture at the same time packets between *ips1* and *tunnel* and save them for later use. Make sure that the *dc0* interface of *ips1* is mirrored to *ohmi*'s *eth1* port.

Unfortunately there is not an IPv6 gateway in the lab hall. Anyway, you can try to define *ips1* as a default router, if you like.

**Question group 5. Report shortly what kind of messages there were in the network when routing was starting. Also compare RIPng to RIPv2 protocol. How is the default route defined with route6d? DO NOT attach captures to your final report.**

After the routing works, find out what are the IPv6 addresses of *pc4* and *pc5*. Remember the *ipv6* command in Windows. Then trace route from *pc5* to *pc4*. Also find out their link level addresses. How has the IPv6 address of *pc4* been formed?

**Question group 6. Tell what IP and link level addresses of *pc5* and *pc3* were. Also tell, if the tunnel has any noticeable effect on the connection? Capture the IPv6 routing table from *ips4* and explain how to interpret it. Attach it to your final report.**

## 5. Anycast and ICMPv6

Now it's time to configure an anycast address to all routers expect to *tunnel*. An anycast address can be any legal address from the (site local) unicast address space. This time use only *ifconfig* to do configurations so you don't have to reboot the router. Again, check the man pages if *ifconfig* to get help.

Ping from *pc4* to the anycast address. Which node answers the ping? Does it answer with its anycast or with unicast address? There are also special anycast addresses to certain services. Mention one service, which has an anycast address.

Try also to ping from *ips1* to *tunnel* and examine ICMPv6 headers. Tell what data the ICMPv6 packets included?

**Question group 7. Answer previous questions?**

## 6. Autoconfiguration

As you have seen the autoconfiguration works well with the PCs. Everything really happens automatically. Now we use autoconfiguration also with routers. Look at the configuration manuals. Keep all your previously configured network addresses same as before, but make the router *tunnel* to configure its host part automatically. Do configurations to *rc.conf* and reboot the machine after changes. You loose now your anycast address, but you won't need it anymore. Remember to start the *route6d* after rebooting.

After you have made the new configurations add the IPv6 address of *tunnel* to */etc/hosts* file in *ips2*. Give the name *airbus* to the address.

**Question group 8. What is the IPv6 address of *tunnel* after autoconfiguration? Tell what changes you did. How did you configure the hosts file?**

## 7. FreeBSD host configurations

Let us disable the link between *ips1* and *ips2*. Make the *ips2* to be a host now, not a router anymore. Use stateless autoconfiguration. Go through the additional material for more help. Trace route from *ips2* to *pc5* to check that the connection works. In *ips2* there is a big file named */root/isotiedosto*. Copy it to *tunnel* with *scp* using both IPv4 and IPv6. (parameters *-4* and *-6*). You can now use the name *airbus* for the IPv6 address of *tunnel*.

**Question group 9. What changes did you make when you configured *ips2* to be a host? What were the average throughputs with IPv4 and with IPv6. Were there any differences? Why?**

## 8. Working with the Faith

You have to configure an IPv6 to IPv4 translator using *faith* daemon. Study the *faithd* manual and the other material you have. Configure the *faithd* to *ips4* and test it from *ips3*. Things that you must do:

1. Write `sysctl -w net.inet6.ip6.keepfaith=1` at *ips4*.
2. Create a virtual *faith* interface with *ifconfig*.
3. Assign a prefix (`3ffe:501:4819:ffff::`) to *faith0* interface and route all traffic to it.
4. Start *faithd* to support *http* protocol.

To test *faith*, try to open a *www* connection to port 80 with *lynx* from *ips3*. Define the address **3ffe:501:4819:ffff::130.233.154.176** to */etc/hosts* file as *confetti* and use it with *lynx*. Where is the connection directed?

Ask the assistant to check that the connection works.

**Question group 10. What kind of problems can *faithd* cause? Does *faithd* work with plain IP traffic? Why?**

**Question group 11. Compare shortly OSPFv2 to OSPFv3?**

## 9. Restore the original settings

Remove all IPv6 configurations from all *rc.conf* files. Leave one model line to each file. Also remove all *tunnel* configurations and the configurations you made for *faith*. Make *ips2* to be a router again.

## 10. Final Report

Please answer question groups 1-11.

## Interface Chart

*Write the names of physical and virtual interfaces to this chart and write also their IPv6 addresses.  
Draw also connections between them.*

### ips1

Interface	IPv6 address
dc0	
dc2	
dc3	
gif0	

### ips2

Interface	IPv6 address
dc0	
dc3	

### ips3

Interface	IPv6 address
dc1	
dc2	

### ips4

Interface	IPv6 address
dc0	
dc1	
dc2	

### tunnel

Interface	IPv6 address
de0	
gif0	