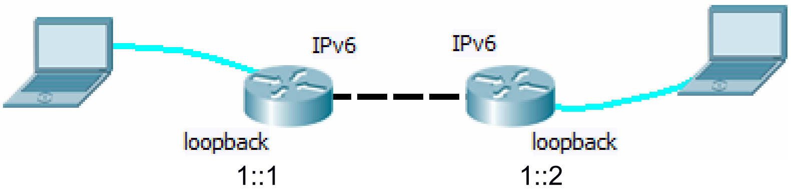
COMPUTER NETWORKS - LABORATORY 076

Subject:
Cisco routers - IPv6.
IPv6 dynamic routing protocols: RIPng, OSPFv3, EIGRP.
IPv6 in IPv4 tunneling.

Task A: Configuring an IPv6 protocol

1. Prepare two Cisco routers having IPv6 functionality (in the lab there are router series of 2600XM, 2800, 2900, 3600, 3700, 3800, 7100, 7200, 7300 or 7400). Connect these router in computer-router-ruter-computer order.
   The connection between the routers should be made using Fast Ethernet or serial interfaces.
2. To make sure the router has an IPv6 functionality, check the command (whether it exists or not):
   Router#show ipv6 ?



3. Examples of special IPv6 addresses:
   - 0:0:0:0:0:0:0:0 - IPv4 of 0.0.0.0 equivalent
   - 0:0:0:0:0:0:0:1 - IPv4 of 127.0.0.1 equivalent

4. Policies related to IPv6 addressing:

   It is possible to define the full contents of the IPv6 address:

   Router(config-if)#ipv6 address 1111:1:0:0:0:0::1111/64

   or just a prefix and host number without an obvious amount of zeroes in the middle:

   Router(config-if)#ipv6 address 1111:1::1111/64

   IPv6 addresses in isolated environment could be quite simple:
   Router(config-if)#ipv6 address 1 :: 1/64

   where :: indicates all the zeros in between.

You can also use an EUI-64 notation that guarantees the uniqueness of IPv6 address:

```
Router(config-if)#ipv6 address 1::/64 eui-64
```

which is assumed to supply only the older part of the address (the prefix), when host ID (64 bits) is provided with a MAC value (L2 address of an interface), in the following order:
MMMM: MMFF: FEMM: MM: MM,
where MM are bytes taken from the MAC, and 0xFFFE is a constant supplement to 64 bits (MAC is 48 in lenght).
Such address can be used with a mask of 64 bits.

Create a loopback interface and assign an EUI-64 IPv6 address to it, then check its contents (generated), e.g.:

```
Router(config)#int loopback 1
Router(config-if)#ipv6 address 5::/64 eui-64
Router(config-if)#no shut
```

Assign a standard IPv6 address as well

```
Router(config-if)#ipv6 address 5::1/64
```

```
Router#show ipv6 int brief
Router#show ipv6 interface loopback 1
```

5. Diagnostics:

```
Router#debug ipv6 icmp
Router#ping ipv6 1::1
```

Configure another router with IPx6 of 1::2 on the Ethernet interface, then check connectivity between these routers

```
Router#ping ipv6 1::2 source 1::1
Router#traceroute ipv6 1::2
```

Enabling debug mode for ICMPv6:

```
Router#show ipv6 route
```

6. In a second router (where there is no information about a route to 5::1 define a static rule (IPv6 routing is similar to IPv4):

```
Router2(config)#ipv6 route 5::/64 1::1
```

Check the connectivity

```
Router2#ping ipv6  5::1
```

Task B: RIPng on IPv6

1. Use a hardware configuration from a previous task, enable IPv6 routing on both routers:

   ```
   Router(config)# ipv6 unicast-routing
   ```

2. *Turn on RIPng routing:*

   ```
   Router(config)# ipv6 router rip ripper
   ```

   where ripper is the name of the RIPng process.

3. IPv6 RIP is process activated by admin on particular interfaces, not networks as it was before in standard RIP for IPv4 case.
   Go to all IPv6 interfaces and enable RIPng there:

   ```
   Router(config)#int fa 0/0
   Router(config-if)#ipv6 rip ripper enable
   Router(config-if)#no shutdown
   Router(config-if)#exit
   Router(config)#int loopback 1
   Router(config-if)#ipv6 rip ripper enable
   Router(config-if)#no shutdown
   ```

4. Remove previously defined IPv6 static rule:

   ```
   Router2(config)#no ipv6 route 5::/64 1::1
   ```

5. Check the configuration and the ability to communicate with IPv6 networking:

   ```
   Router#show ipv6 int fa 0/0
   Router#show ipv6 route
   Router#show ipv6 rip

   Router2#ping ipv6  5::1
   ```

   Refreshing a routing table contents after some changes:

   ```
   Router#clear ipv6 route *
   ```

Task C: OSPFv3 and EIGRP IPv6 routing

1. Similarly, it is possible to use EIGRP and OSPFv3 on IPv6. Disable RIPng and start OSPFv3 instead:

   ```
   Router(config)#no ipv6 router rip ripper
   Router(config)#ipv6 router ospf 10
   ```

   where *10* is OSPF process identifier.
   For OSPFv3 it is necessary to manually configure the OSPF router-id (for all OSPF routers involved). It cannot be generated with a use of IPv6 addresses, like it was in an IPv4 case). The value of the router-id is 32 bit long must be unique. Define it manually, different value for each router:

```
Router(config-router)#router-id 1.2.3.4
```

In OSPFv3 the OSPF process of is also assigned do interfaces, instead networks (and thus - the networks directly connected):

```
Router(config)#interface fa 0/0
Router(config-if)#ipv6 ospf 10 area 0.0.0.0
```

Create OSPFv3 network composed of at least two routers and check the functioning:

```
Router#debug ipv6 ospf
Router#debug ipv6 ospf packet
Router#debug ipv6 ospf hello
```

```
Router#show ipv6 protocols
Router#show ipv6 route
Router#show ipv6 ospf neighbor
Router#show ipv6 ospf interface
Router#show ipv6 ospf database
```

2. In the opposite router create additional loopback interface, add it to OSPFv3 but in another area (creating an ABR router in OSPFv3):

   ```
   Router2(config)#int loopback 2
   Router2(config-if)#ipv6 address 2::1/64
   Router2(config-if)#no shut
   Router2(config-if)#ipv6 ospf area 1
   ```

   Then check the current known routes base in a first router:

   ```
   Router#sh ipv6 ospf border-routers
   ```

3. Starting an EIGRP for IPv6:
   Remark: EIGRP for IPv6 is implemented only in routers designed for network backbone - in the Laboratory these are Cisco 3640, 3660, 3725, 3745, 3845, 7100, 7200VXR, 7300, 7400, 6500. If you have such routers, run EIGRP for IPv6 (disabling OSPFv3 first):

   ```
   Router(config)#no IPv6 ospf router 10
   Router(config)# ipv6 router eigrp 1234
   ```

   where *1234* is EIGRP identifier of autonomous system (common to all routers in the system).
   EIGRP for IPv6 also needs to have a router ID defined in each router. In some situations (when it is not generated automatically), it is necessary to define id manually. Router-id values must be unique:

   ```
   Router (config) # IPv6 router EIGRP 1234
   Router (config-rtr) #eigrp router-id 10
   ```

   Like OSPFv3, EIRGP must be assigned to all interfaces involved in the EIGRP routing process:

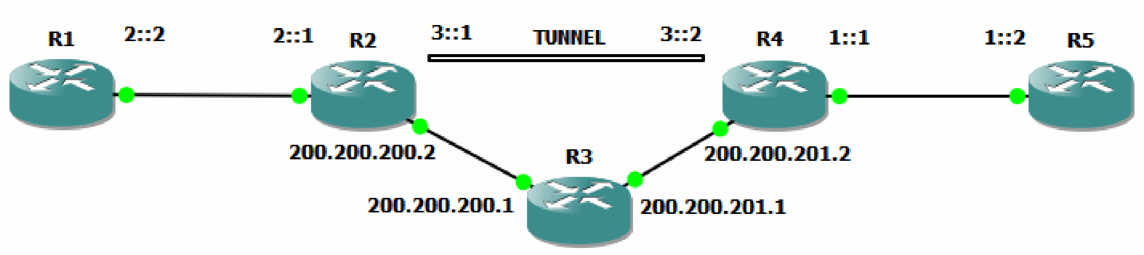
   ```
   Router(config)#int fa 0/0
   Router(config-if)# ipv6 eigrp 10
   ```

   After running EIGRP check its operation (similarly to OSPFv3).

Task D: IPv6 tunneling over IPv4

IPv6IP tunneling allows to transmit IPv6 traffic over non-IPv6 routers, between IPv6 networks. It uses a typical tunneling process with encapsulation of IPv6 traffic in IPv4.
The current task can be performed together with another team.
To create a full scale system there are five routers needed(R1 - R5). In the absence of a sufficient number of IPv6 routers R1 and R5 can be omitted and simulated by loopback test interfaces on routers R2 and R4.

Build a network topologys it is shown in a diagram below
IPv6IP tunnel will defined between routers R2 and R4. Router R3 will transfer only IPv4 packets, not interfering with IPv6 addressing:



1. Prepare the configuration of routers R1 and R5. These routers must be set to support RIPng for IPv6:

Router R1:

```
ipv6 unicast-routing
ipv6 router rip ripper
interface FastEthernet 0/0
 no ip address
 ipv6 address 2::2/64
 ipv6 rip ripper enable
 no sh
exit
```

Router R5:

```
ipv6 unicast-routing
ipv6 router rip ripper
interface FastEthernet 0/0
 no ip address
 ipv6 address 1::2/64
 ipv6 rip ripper enable
 no sh
exit
```

Comment: *ipv6 unicast-routing* command is used to enable IPv6 routing (default is off)

2. Prepare configuration of gateway routers (R2 and R4):
Configure and enable IPv6 tunnel interfaces , IPv4 interfaces (Fast Ethernets 0/0) and “local” IPv6 interfaces (Fast Ethernets 0/0).
Set tunnel mode to IPv6IP
Enable RIPng routing on all IPv6 interfaces
Enable OSPF routing on all IPv4 interfaces

Complete configuration:

Router R2:

```
ipv6 unicast-routing
ipv6 router rip ripper

interface FastEthernet 0/0
 ip address 200.200.200.2 255.255.255.0
 no sh
exit
interface FastEthernet 0/1
 no ip address
 ipv6 address 2::1/64
 ipv6 rip ripper enable
 no sh
exit
router ospf 1
 log-adjacency-changes
 network 200.200.200.0 0.0.0.255 area 0
 router-id 200.200.200.2
exit
interface Tunnel0
 no ip address
 ipv6 address  3::1/64
 ipv6 rip ripper enable
 tunnel source FastEthernet 0/0
 tunnel destination 200.200.201.2
 tunnel mode ipv6ip
 no sh
exit
```

Router R4:

```
ipv6 unicast-routing
ipv6 router rip ripper

interface FastEthernet 0/0
 ip address 200.200.201.2 255.255.255.0
 no sh
exit
interface FastEthernet 0/1
 no ip address
 ipv6 address 1::1/64
 ipv6 rip ripper enable
 no sh
exit
router ospf 1
 log-adjacency-changes
 network 200.200.201.0 0.0.0.255 area 0
 router-id 200.200.201.2
exit
interface Tunnel0
 no ip address
 ipv6 address 3::2/64
 ipv6 rip ripper enable
 tunnel source FastEthernet 0/0
 tunnel destination 200.200.200.2
 tunnel mode ipv6ip
 no sh
exit
```

3. Prepare configuration for router R3. The router should only handle IPv4 traffic and OSPF:

   Router R3:

```
ip subnet-zero
ip classless

interface FastEthernet 0/0
 ip address 200.200.200.1 255.255.255.0
 no sh
exit
interface FastEthernet 0/1
 ip address 200.200.201.1 255.255.255.0
 no sh
exit

router ospf 1
 log-adjacency-changes
 network 200.200.200.0 0.0.0.255 area 0
 network 200.200.201.0 0.0.0.255 area 0
 router-id 200.200.201.1
exit
```

4. Check the status of all interfaces in all routers respectively:

```
Router#show ip int brief
Router#show ipv6 int brief
```

Check the operation of all OSPF routers (operating in IPv4):

```
Router#show ip ospf neighbor
```

5. Verify the contents of routing tables in routers R2, R3, R4:

```
Router2#show ip route
Router2#show ipv6 route
Router3#show ip route
Router3#show ipv6 route
Router4#show ip route
Router4#show ipv6 route
```

Check the communication in IPv6 infrastructure (ping and traceroute with ranges increasing) e.g.:

```
Router1#ping ipv6 2::2
Router1#ping ipv6 2::1
Router1#ping ipv6 3::2
Router1#ping ipv6 2::2
Router1#ping ipv6 3::2
Router1#ping ipv6 1::1
Router1#ping ipv6 1::2
```