universidade de aveiro
theoria poiesis praxis

Fundamentos de Redes

Objectives

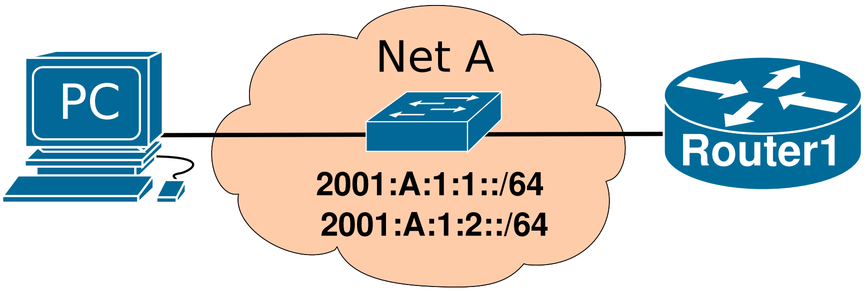
- IPv6 addressing
- Configuration and analysis of IPv4 and IPv6 static routing.
- Familiarization with IP routing protocols.
- Configuration and analysis of the RIPv1, RIPv2 and RIPng protocols.

Duration

- 3 weeks

For this point on, use GNS3 to assemble the networks and deploy the protocols/mechanisms.

IPv6 Basic Mechanisms



1. Considering the above depicted network, start by connecting the PC (a VirtualBox VM Linux) to the switch without any other connections.
To avoid incompatibilities, disable the Linux network manager (if active):

```
sudo service network-manager stop
```

Note: The commands `sudo service network-manager start` can be used to restart the application/service.
Start a capture in the link between the PC and the Switch. Turn off and on the PC's Ethernet interface:

```
sudo ifconfig eth0 down
sudo ifconfig eth0 up
```

Stop the capture and analyze the IPv6 packets.

2. Connect Router1 to the switch and start a capture in the link between the PC and the Switch. Power on Router1 and configure it's interface to network A.

```
Router1(config)# ipv6 unicast-routing
Router1(config)# interface <if-name>
Router1(config-if)# ipv6 enable
Router1(config-if)# no shutdown
```

Verify router's interfaces names and configuration:

```
Router1# show ipv6 interface
Router1# show ipv6 interface brief
```

Restart PC's Ethernet interface and verify it's interface information:

```
sudo ifconfig eth0 down            (sudo ifconfig enp4s0 down)
sudo ifconfig eth0 up              (sudo ifconfig enp4s0 up)
ifconfig eth0                      (ifconfig enp4s0)
```

Stop the capture and analyze the IPv6 packets and equipment’s information. Use the commands:

```
show ipv6 interface brief
show ipv6 route
```

to verify interfaces' IPv6 addressing and verify router's IPv6 routing table.

3. Re-start a capture in the link between the PC and the Switch. Configure Router's interface with a manually defined IPv6 global address from network 2001:A:1:1::/64.

```
Router1(config)# interface <if-name>
Router1(config-if)# ipv6 address 2001:A:1:1::100/64
Router1(config-if)# no shutdown
```

Verify PC's Ethernet interface information. Stop the capture and analyze the IPv6 packets. Verify Router's interfaces IPv6 addresses and router's IPv6 routing table.
>> Explain the process by which the PC obtained the IPv6 addresses.

4 Re-start a capture in the link between the PC and the Switch. Configure Router's interface with a EUI-64 based IPv6 global address from network 2001:A:1:2::/64.

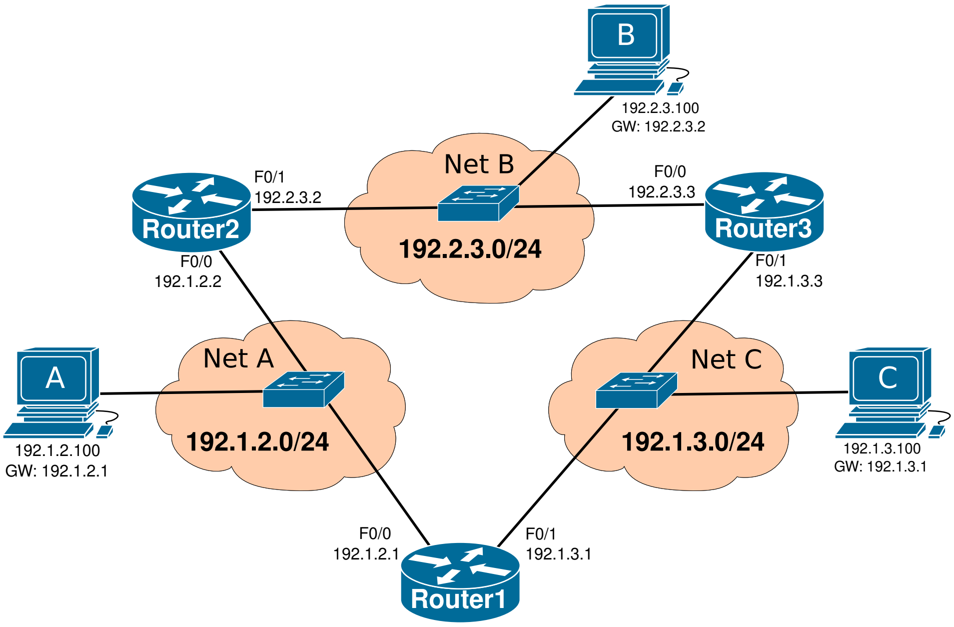
```
Router1(config)# interface <if-name>
Router1(config-if)# ipv6 address 2001:A:1:2::/64 eui-64
Router1(config-if)# no shutdown
```

Verify PC's Ethernet interface information. Stop the capture and analyze the IPv6 packets. Verify Router's interfaces IPv6 addressing and the router's IPv6 routing table
>> Explain the process by which the Router completed the last 64 bits of its IPv6 addresses.

>> Discuss a possible disadvantage of using the standard EUI-64 at routers' interfaces.
>> Does the process by which the PC obtained the IPv6 addresses, changed by using the EUI-64 standard at the Router.

5. <u>Re-start a capture in the link between the PC and the Switch.</u> At the PC, using the command *ping6* perform a ping to:
 a) Router's Link-Local address (you need to define the output interface with option "`-I eth0`" or "`-I enp4s0`").
 b) Router's Global address from network 2001:A:1:1::/64.
 c) Router's Global address from network 2001:A:1:2::/64.
Stop the capture and analyze the IPv6/ICMPv6 packets.
>> Explain the physical addresses resolution process in IPv6.

IPv4 Static Routing



6.Assemble the above depicted network **<u>using GNS3</u>** and configure all IPv4 addresses. Verify the interfaces' configurations, routing tables, and IPv4 ARP tables:

```
Router1# show ip interface brief
Router1# show ip route
Router1# show ip arp
```

<u>Start a packet capture in all networks (PC-SW connection)</u>, and execute multiple ping commands to test the connectivity between the different equipment's interfaces.
>> Explain the reasons to have connectivity between some interfaces and not others.
>> Explain the reasons why some *ICMP Requests* reach the destination network but the ping is not successful.
>> Explain why *ICMP Destination Unreachable* packets are sent sometimes by the routers.

7. Configure at Router1 a static route to network B via Router2:

```
Router1(config)# ip route 192.2.3.0 255.255.255.0 192.1.2.2
```

Execute a ping command on PC A for the 3 IPv4 addresses of network B.
>> Justify the resulting routing tables on all routers.
>> Explain the reason why some IPv4 addresses can be reached and others not.
>> Explain why an *ICMP Redirect* packet is sometimes sent by Router 1.

8. Configure one static route on Router 2 and another one on Router 3 in order to obtain full connectivity. Register and justify the resulting routing tables of all routers. With the *ping* command, verify that all PCs have connectivity with all IPv4 addresses.

9. Disable (with command `shutdown`) the interface of Router 2 with network B (simulating, in this way, an interface failure). Register and justify the routing table of Router 1. Execute a ping command from Router 1 to PC B.
>> Explain the main disadvantage of using static routes.

10. **Restore the disconnected interface and remove all static routes** configured in the previous experiments (`no ip route *`).

IPv4 Dynamic Routing with RIPv1

11. Configure the RIPv1 protocol at all routers. For Router 1:

```
Router1(config)# router rip
Router1(config-router)# version 1
Router1(config-router)# network 192.1.2.0
Router1(config-router)# network 192.1.3.0
```

Repeat similar configurations on Routers 2 and 3. Verify that the network has full connectivity.
>> Verify and justify the routing tables obtained at all routers.

12. **Start a packet capture in all networks (PC-SW connection)** and captured at least 6 RIP packets in each network. Analyze the captured packets
>> Determine the type of RIP packets sent by each router and its periodicity.
>> Determine the IPv4 destination address, transport protocol and port numbers used by RIPv1.
>> Determine if *split-horizon* is being used or not (default setting).
>> Explain how RIPv1 works.

13. Invert the *split-horizon* configuration at the routers' interfaces that are connected to network A.
To turn *split-horizon* on an interface:

```
Router1(config)# interface F0/0
Router1(config-if)# ip split-horizon
```

To turn *split-horizon* off on an interface:

```
Router1(config)# interface F0/0
Router1(config-if)# no ip split-horizon
```

Capture again at least 6 packets. Analyze the captured packets and compare their content with the packets captured in the previous experiment.
>> Justify the content of RIPv1 packets when *split-horizon* is used and when it is not used.
Restore the *split-horizon* default configuration at the router interfaces connected to network A.

14. **Start a packet capture in all networks (one of the Router-SW connections).** Execute ping commands from Router 1 to all IPv4 addresses of network B. From the packet captures, infer the routing paths followed by the ICMP packets after each ping command.
>> Explain why the packets (from different ping commands) followed different paths.

15. **Start a packet capture in network A (Router 1-SW connection).** Disable (with command `shutdown`) the interface of Router 1 with network C
>> Register and justify the routing tables obtained at all routers.
>> By analyzing the captured packets, explain how RIP protocol propagated the information after the failure.

16. **Start a packet capture in network A (Router 1-SW connection).** Re-enable (with command `no shutdown`) the interface of Router 1 with network C
>> Register and justify the routing tables obtained at all routers.
>> By analyzing the captured packets, explain how RIP protocol propagated the information after the failure recovery.

17. Configure at Router1 a static route to network B via Router2:

```
Router1(config)# ip route 192.2.3.0 255.255.255.0 192.1.2.2
```

>> Register and justify the routing tables obtained at all routers.
>> Explain why Router 1 does not have any routes to network B learned with RIP.

18. Re-configure at Router1 the static route to network B via Router2 to have an administrate distance of 200:

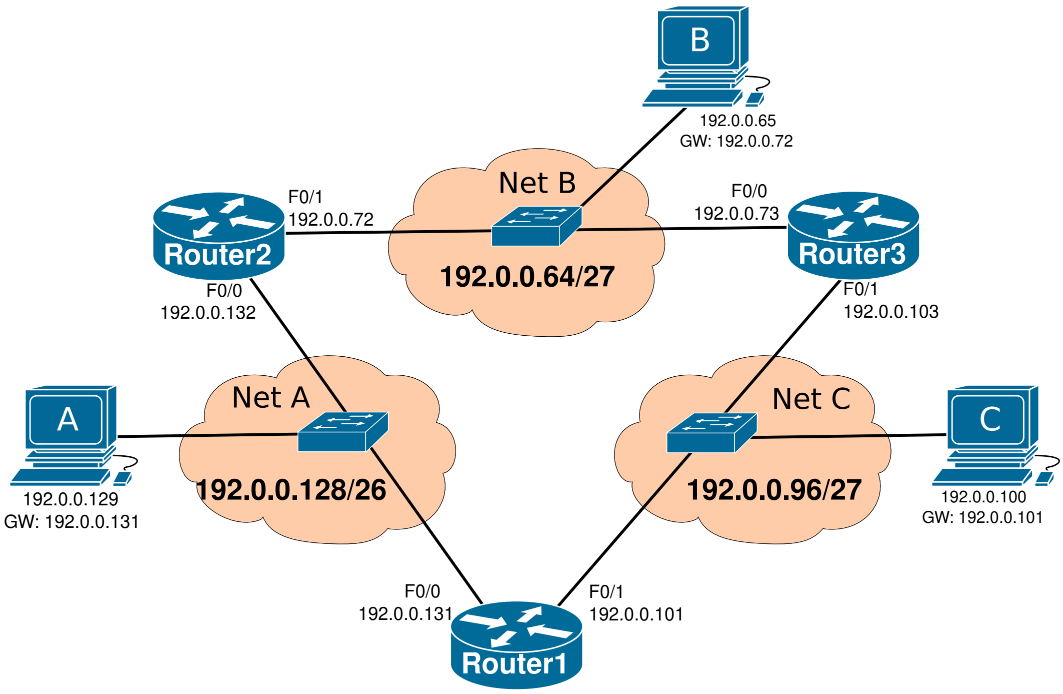
```
Router1(config)# no ip route 192.2.3.0 255.255.255.0 192.1.2.2
Router1(config)# ip route 192.2.3.0 255.255.255.0 192.1.2.2 200
```

>> Register and justify the routing tables obtained at all routers.
>> Explain how the administrative distance works.

19. Disable the RIP processes on all routers (with the command `no router rip`) and delete the static route at Router 1.

IPv4 Dynamic Routing with RIPv2

<u>Remember that</u> RIPv2 adds three features to RIPv1:
(i) it includes the network masks in the distance vector information exchanged in the RIP Response messages, supporting in this way Variable Length Subnet Masking;
(ii) RIP messages are sent to the IP multicast address 224.0.0.9 (instead the broadcast address 255.255.255.255), avoiding hosts to process the RIP messages;
(iii) it adds authentication to the exchange of RIP messages, preventing third party entities to change the routing tables.



20. Assuming that that the network manager has only the 192.0.0.0/24 network prefix (a class C network) to use on its network, it decides to use subnets 192.0.0.128/26, 192.0.0.64/27, and 192.0.0.96/27. Reconfigure all IPv4 addresses according to the above figure, and reconfigure a RIPv1 process in all Routers.

```
Router1(config)# router rip
Router1(config-router)# version 1
Router1(config-router)# network 192.0.0.0
Router1(config)# interface F0/0
Router1(config-if)# ip rip receive version 1
Router1(config-if)# interface F0/1
Router1(config-if)# ip rip receive version 1
```

Note: routers must be forced to receive RIPv1 packets in an interface connected to a subnet with the command `ip rip receive version`.
Repeat similar configurations on Routers 2 and 3. Verify if the network has full connectivity.
>> Register and justify the routing tables obtained at all routers.
>> Explain why RIPv1 does not work properly in this case.

21. Configure the RIPv2 protocol at all routers. For Router 1:

```
Router1(config)# router rip
Router1(config-router)# version 2
Router1(config-router)# network 192.0.0.0
Router1(config)# interface F0/0
Router1(config-if)# no ip rip receive version 1
Router1(config-if)# interface F0/1
Router1(config-if)# no ip rip receive version 1
```

Repeat similar configurations on Routers 2 and 3.
>> Verify that the network has full connectivity.
>> Verify and justify the routing tables obtained at all routers.

22. **Start a packet capture in all networks (PC-SW connection)** and captured at least 6 RIP packets in each network. Analyze the captured packets
>> Determine the type of RIP packets sent by each router and its periodicity.
>> Determine the IPv4 destination address, transport protocol and port numbers used by RIPv2.
>> Explain how RIPv2 works.

23. Disable the *split-horizon* configuration at the routers' interfaces that are connected to network A.
To turn *split-horizon* on an interface:

```
Router1(config)# interface F0/0
Router1(config-if)# ip split-horizon
```

To turn *split-horizon* off on an interface:

```
Router1(config)# interface F0/0
Router1(config-if)# no ip split-horizon
```

Capture again at least 6 packets. Analyze the captured packets and compare their content with the packets captured in the previous experiment.
>> Justify the content of RIPv2 packets when *split-horizon* is used and when it is not used.
Restore the *split-horizon* default configuration at the router interfaces connected to network A.

24. Activate RIPv2 MD5 Authentication mode (with password *labcom*) in both interfaces of Router 1:

```
Router1(config)# key chain key_chain
Router1(config-keychain)# key 1
Router1(config-keychain-key)# key-string labcom
Router1(config)# interface range F0/0 - 1
Router1(config-if-range)# ip rip authentication key-chain key_chain
Router1(config-if-range)# ip rip authentication mode md5
```

>> Register and justify the routing tables obtained at all routers.

25. **Start a packet capture in all networks (PC-SW connection)** and captured at least 6 RIP packets in networks A and C. Analyze the captured packets
>> Analise the RIPv2 packets sent by Router 1, namely the authentication data field.

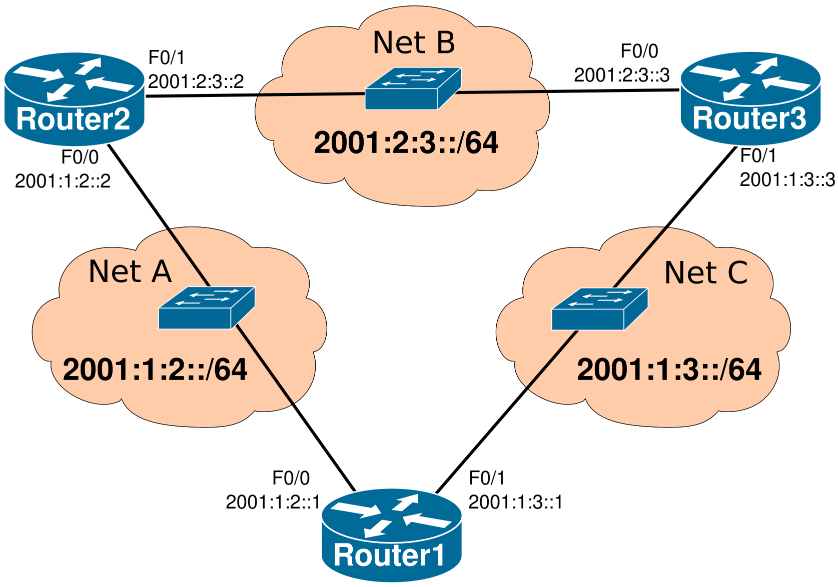
26. Activate RIPv2 MD5 Authentication mode (with password *labcom*) in the other Routers' interfaces:
>> Register and justify the routing tables obtained at all routers.
>> Justify the importance of using authentication.

27. **Start a packet capture in all networks (PC-SW connection)** and captured at least 6 RIP packets in each network. Assuming that Router 1 provides Internet connectivity, configure it to announce a default route via RIPv2:

```
Router1(config)# router rip
Router1(config-router)# default-information originate
```

>> Register and justify the routing tables obtained at all routers.
>> Explain how an IPv4 packet destined to a network not known by the routers is routed.
>> Analyze the captured RIPv2 packets and explain how the default route is announced.

IPv6 Static Routing



28. Configure all IPv6 addresses and activate IPv6 routing with the command:

```
Router1(config)# ipv6 unicast-routing
```

Verify the interfaces' IPv6 addresses and status, routing tables, IPv6 neighbors and running IPv6 protocols:

```
Router1# show ipv6 interface brief
Router1# show ipv6 route
Router1# show ipv6 neighbors
Router1# show ipv6 protocols
```

Start a packet capture in all networks (Router-SW connection), and execute multiple ping commands to test the connectivity between the different equipment's interfaces.
>> Explain the reasons to have connectivity between some interfaces and not others.

29. Configure all necessary static routes in order to achieve full IPv6 connectivity:

```
Router(config)# ipv6 route <ipv6-net> <ipv6_next_hop>
```

>> Reverify the routing tables and retest the connectivity between the equipments.

IPv6 Dynamic Routing with RIPng

30. Remove all static routes that were configured in the previous experiment.
Activate the RIPng protocol (process named proc1) in all routers interfaces:

```
Router1(config)# interface range F0/0 - 1
Router1(config-if-range)# ipv6 rip proc1 enable
```

Repeat similar configurations on Routers 2 and 3.
Verify also the RIPng information in routers:

```
show ipv6 rip proc1
show ipv6 rip proc1 database
show ipv6 rip proc1 next-hops
```

>> Re-verify the routing tables and retest the connectivity between the equipments.

31. **Start a packet capture in all networks (Router-SW connection)** and captured at least 6 RIPng packets in each network. Analyze the captured packets
>> Determine the type of RIPng packets sent by each router and its periodicity.
>> Determine the IPv6 source and destination addresses, transport protocol and port numbers used by RIPng.
>> Explain how RIPng works.

32. **Start a packet capture in all networks (Router-SW connection)** and disable the *split-horizon* configuration in Router 1.
To turn *split-horizon* off:

```
Router1(config)# ipv6 router rip proc1
Router1(config-rtr)# no split-horizon
```

Capture again at least 6 packets. Analyze the captur.d packets and compare their content with the packets captured in the previous experiment.
>> Justify the content of RIPng packets when *split-horizon* is used and when it is not used.
Restore the *split-horizon* default configuration in Router 1.

33. **Start a packet capture in all networks (Router-SW connection)** and captured at least 6 RIPng packets in each network. Assuming that Router 1 provides Internet connectivity, configure it to announce a default route via RIPng:

```
Router1(config)# interface range F0/0 - 1
Router1(config-if-range)# ipv6 rip proc1 default-information originate
```

>> Register and justify the routing tables obtained at all routers.
>> Explain how an IPv6 packet destined to a network not known by the routers is routed.
>> Analyze the captured RIPng packets and explain how the default route is announced

Extra – IPv4 and IPv6 Dynamic Routing with OSPFv2 and OSPFv3

34. Activate OSPFv2 and OSPFv3 protocols (process number 1, single area) in all routers interfaces:

```
Router1(config)# interface F0/0
Router1(config-if)# ip ospf 1 area 0
Router1(config-if)# ipv6 ospf 1 area 0
```

Repeat similar configurations on other interfaces and Routers 2 and 3.
Analyse also the OSPFv2 and OSPFv3 information in routers:

```
show ip ospf database                    !for a database summary
show ip ospf database router             !for the  Router Link States
show ip ospf database network            !for Network Link States
show ipv6 ospf 1 database
show ipv6 ospf 1 database network
show ipv6 ospf 1 database router
show ipv6 ospf 1 database prefix         !New OSPFv3 database
```

>> Re-verify the routing tables and retest the connectivity between the equipments.

35. Configure Router 1 to announce a default IPv4 route:

```
Router1(config)# router ospf 1
Router1(config-router)# default-information originate always
```

>> Re-verify the routing tables and identify the default route being announced by Router1.