



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI



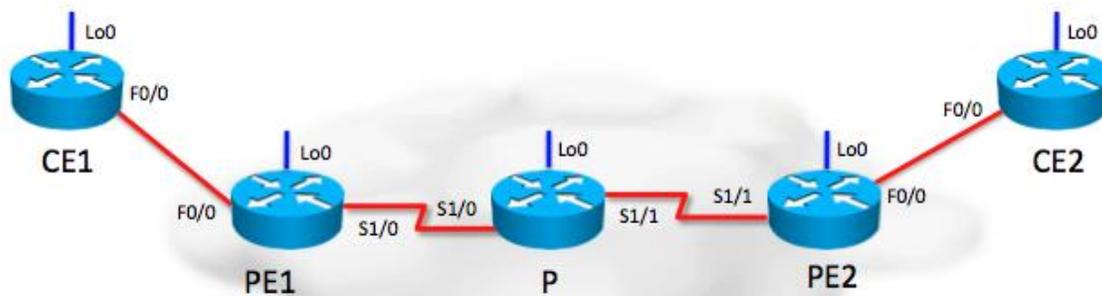
Instrumente Structurale  
2007-2013



Platformă de e-learning și curriculumă  
e-content pentru învățământul superior tehnic

Securizarea Calculatoarelor și a Rețelelor

**34. Implementarea unei rețele MPLS**



## IP addressing table

Device	Interface	IP address	Subnet mask
CE1	Loopback0	199.0.0.1	255.255.255.255
	FastEthernet0/0	10.1.1.1	255.255.255.252
PE1	Loopback0	150.0.1.1	255.255.255.255
	FastEthernet0/0	10.1.1.2	255.255.255.252
	Serial1/0	192.168.1.1	255.255.255.252
P	Loopback0	150.0.1.2	255.255.255.255
	Serial1/0	192.168.1.2	255.255.255.252
	Serial1/1	192.168.1.5	255.255.255.252
PE2	Loopback0	150.0.1.3	255.255.255.255
	Serial1/1	192.168.1.6	255.255.255.252
	FastEthernet0/0	10.1.1.6	255.255.255.252
CE2	Loopback0	99.0.0.1	255.255.255.255

	FastEthernet0/0	10.1.1.5	255.255.255.252
--	-----------------	----------	-----------------

## Short tutorials

**Routing protocols** are protocols that transport routes between routers. This is an automated alternative to having to manually configure all required static routes.

Some routing protocols do not inform the router about the entire network topology, but instead they only know about destinations learned from their directly connected neighbors and how far these destinations are. Usually, these routing protocols cause routers to periodically send their entire routing tables to all their neighbors. When such a router receives another router's routing table, it will only know that those destinations are accessible through the neighbor that sent them, without any knowledge of the network topology. These routing protocols are called **distance-vector routing protocols** (because they only know the distance and the direction to go in order to reach a destination).

Other routing protocols keep a topological database of the entire network. These are usually more processor and memory intensive for the router, but they react much more quickly to network changes (providing more uptime) and they use more efficient algorithms to calculate the best paths. These are called **link-state routing protocols**.

**Administrative distance:** a value that indicates the preference for a route based on how it was originated. A route to a certain destination can be obtained from multiple sources (different routing protocols, static routes, connected routes). Each source has a different administrative distance and the router will prefer the lowest value.

Directly connected routes have an AD of 0.

Static routes have an AD of 1.

Routing protocols have different ADs: 90 (EIGRP), 110 (OSPF), 115 (IS-IS), 120 (RIP) and many others.

**Metric:** a value that indicates the „best“ route to a destination when multiple routes from the same source are available. Metrics are used to decide which route will the router use when (for example) the same routing protocol provides two or more routes to the same destination.

This happens when there are redundant paths in the network. Each routing has its own method for calculating the metric of its routes. Some metrics are as simple as counting the number of routers (hops) traversed towards the destination, others take into consideration bandwidths, delays and other parameters. Lowest metric values are considered to indicate better routes.

## 1.1 Quick RIP tutorial

RIP is a simple distance-vector routing protocol suited for small networks. Configuring RIP is very simple. The following command enables RIP configuration mode:

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

It is strongly recommended to enable RIP version 2 and disable auto summarization to get rid of RIPv1's default classful behaviour:

```
Router(config-router)# version 2
Router(config-router)# no auto-summary
```

The „network” command in RIP does two things: selects which interfaces will participate in RIP and which networks will be advertised to neighbors:

```
Router(config-router) network 10.0.0.0
Router(config-router) network 192.168.1.0
```

The „network” command is classful. The network address following the „network” command is considered to have its classful network address (/8 for A, /16 for B, /24 for C). The router looks at its active interfaces and checks whether any IP addresses configured on these interfaces fit in the network address of a „network” statement.

For example, if a router has the following interfaces:

```
Fa0/0 – 192.168.1.1/24
S2/1 – 172.16.78.1/30
S2/2 – 172.16.192.1/30
```

Only two „network” statements would be required to activate RIP on these three interfaces:

```
Router(config-router) network 192.168.1.0
Router(config-router) network 172.16.0.0
```

Use „show ip protocols” to verify RIP configuration and „show ip route” to see RIP-learned routes in the routing table (lines that begin with an „R”).

## 1.2 Quick OSPF tutorial

OSPF is much „smarter” than RIP but we won't go into detail here. Configuring OSPF is somewhat similar to RIP, with a few exceptions. Firstly, OSPF can run multiple processes on the

same router so when you enter OSPF configuration mode you must specify which process you are configuring (you should only use one process pretty much all the time).

```
Router(config)# router ospf 1
```

Afterwards, you need to enter the „network” statements again. Here is a sample OSPF „network” statement:

```
Router(config-router)# network 192.168.1.0 0.0.0.255 area 0
```

OSPF „network” statements are not classful, like RIP’s, so you will need a network IP address and a wildcard mask. To keep things simple, a wildcard mask is a negated version of the network mask. Here are some examples:

Network mask: /24 = 255.255.255.0; Wildcard mask: 0.0.0.255

Network mask: /12 = 255.240.0.0; Wildcard mask: 0.15.255.255

Finally, the „area 0” parameter is used because OSPF can support multiple areas inside the same domain. We won’t go into any details here about multiarea OSPF, so keep in mind to always use „area 0” when configuring OSPF in this lab.

Use „show ip protocols” to verify OSPF configuration and „show ip route” to view OSPF routes in the routing table (lines that begin with an „O”).