Computer Networks (*Redes de Computadores - RCOMP*) – 2023/2024

Laboratory Class 06 (PL06 – 3 hours)

Layer three redundancy. Dynamic routing with RIP, EIGRP and OSPF. Autonomous systems. Autonomous systems interconnection – routes redistribution. Cisco IOS Telephony Services (ITS). Packet Tracer activities.

1. Dynamic routing

In a set of interconnected IPv4 networks there may be alternative paths for packets to travel between two IPv4 networks. This is a good thing; it can be used to provide fault tolerance and traffic load balancing.

Yet, if static routing tables are used, routers will not be able to take advantage of alternative paths. This is because if routing tables are static, traffic always follow the same path. And this will happen even if there's a broken link on the used path and an alternative path is available, or even if the used path is highly congested and there is an idle alternative path.

By opposition, **dynamic routing** means routing tables will be always changing to reflect the network infrastructure situation. Creating and keeping updated dynamic routing tables on routers along the network infrastructure is what **routing protocols** have been designed for.

With alternative paths, two columns routing tables become insufficient. In the routing table presented in Table 1, there are three alternative next hops to reach network 192.160.20.0/24.

Table 1 - Routing table with alternative paths

Destination	Next hop
192.168.20.0/24	172.18.30.2
192.168.20.0/24	10.30.42.11
192.168.20.0/24	172.30.104.151

When alternative paths exist, some routers' routing tables will have more than one line for the same destination (network), each with a different next-hop. There must be a criterion for selecting the best alternative, that criterion is called the path metric or cost (Table 2). Smaller metric value or cost means the best choice.

Table 2 - Routing table with the Metric/Cost column

Destination	*Next hop*	**Metric/Cost**
192.168.20.0/24	172.18.30.2	**4**
192.168.20.0/24	10.30.42.11	**8**
192.168.20.0/24	172.30.104.151	**2**

Each routing protocol has its own metric calculation algorithm; therefore, different protocols' metric values cannot be compared. However, within the same routing protocol, the best path always corresponds to the lower metric value. In the above table traffic to network 192.168.20.0/20 would be sent to next-hop 172.30.104.151.

If due to some network problem, the 172.30.104.151 neighbour router becomes unreachable, the routing protocol removes the line from the routing table, thus, from that instant on, the 172.13.30.2 next hop would be used instead.

The elapsed time, since something changes on the network, until the routing protocol is able to reflect that on all routing tables, is called the **convergence time**.

2. Routing Information Protocol (RIP)

RIP is a rather simple and a rather old dynamic routing protocol. To build routing tables, RIP uses a distance-vector algorithm. The main idea around distance-vector algorithms is routing tables are gradually built while routing tables are transmitted between neighbour routers. Simply said, this means received routing tables are merged with the already existing local routing table in local the router.

The RIP metric is the number of routers required to be crossed to reach the destination network, this can also be called the number of hops. This happens because with RIP, usually each link (router interface) is assigned with a cost one metric, and when a routing table is received from a neighbour router the link's cost is added to each rule path metric.

The original RIP standard is at present called RIP version one (RIPv1). Because RIPv1 messages don't carry the network prefix-length, it can only be used in infrastructures using classful IPv4 addressing.

RIPv2 overcomes this issue and supports classless addressing (CIDR – Classless Inter-Domain Routing). RIPv2 also spreads information to neighbour routers by using multicast, instead of broadcast as used by RIPv1.

Both RIPv1 and RIPv2 have a scaling limit problem, they cannot cope with metric values over 15, in fact, metric value 16 is used to represent an unreachable destination. This, of course, limits the use of RIP to network infrastructures only up to some dimension. Even if, as usual in RIP, all interfaces have cost one, there cannot be more than 15 routers between two networks.

This problem is worsened due to the fact RIP doesn't support autonomous systems identifiers. Thus, dividing the infrastructure into several RIP autonomous systems is not straightforward. The issue here is an Autonomous System Boundary Router (ASBR) can use RIP on only one of the connected autonomous systems as we will see on the next lesson.

2.1. RIP practical exercise. Create the Cisco Packet Tracer diagram shown in Figure 1

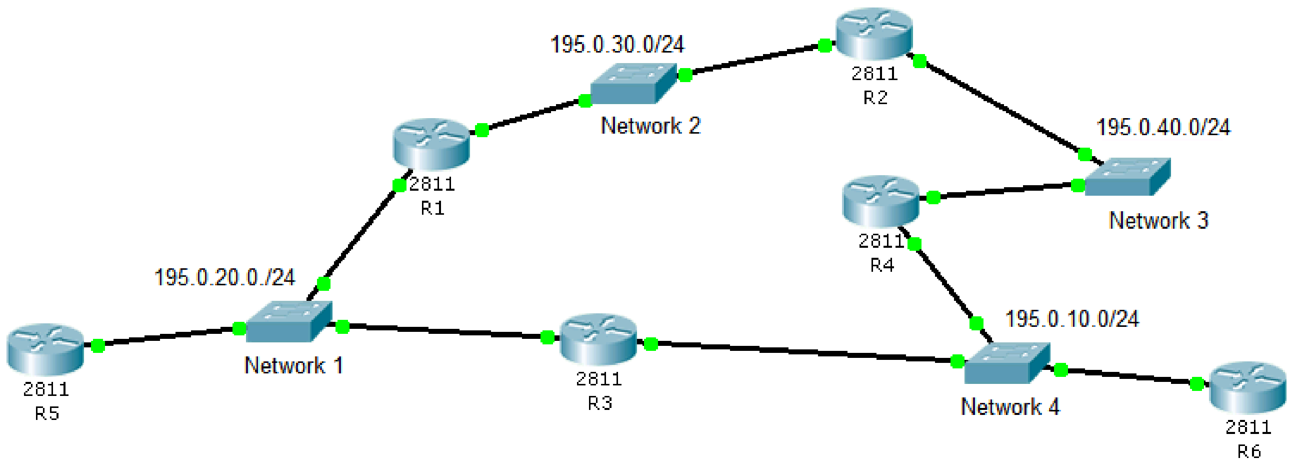

Figure 1 - Networks layout with redundant paths - RIP

a) Assign to every router (R1 to R6) valid IPv4 addresses in each connected network.

b) For every router, activate RIP in all interfaces.

RIP may be configured by using the Packet Tracer graphical forms interface.

If we wait a while (convergence time) all routing tables will be filled. We can use the Inspect tool (magnifying glass) to view each router routing table, lines added by RIP are marked as being type R.

Overall, we have four networks, routers R1, R2, R3, and R4 are connected to two networks, so their routing tables are expected to have two additional lines added by RIP, check that with the inspect tool.

For routers R5 and R6, because each is connected to a single network, we would expect three lines added by RIP. However, if you check it, you will see **4 lines marked as type R and not 3**. Take a closer look to these two routing tables and try to explain why this is happening.

Remember, the routing tables we are seeing have already been optimized by the router, they only contain the best option (lower metric) for each destination, and thus higher metric alternatives are automatically removed.

c) In simulation mode send an ICMP echo request from R5 to R6.

Check the path followed by ICMP packets.

d) Shutdown one of R3 network interfaces.

This is going to make it impossible for packets to follow the earlier path.

Now, we must wait for the convergence to take place, you'd better be in real mode, otherwise, it will take quite a long time. Use again the Inspect tool to know when routing tables have been changed by RIP and are stable again.

e) In simulation mode, again, send and ICMP echo request from R5 to R6 to check through where packets are traveling now.

f) Now let's go back to the starting point. Enable again the shutdown interface in R3.

Wait for convergence, and check that routing tables are now restored to the first state.

3. EIGRP (Enhanced Interior Gateway Routing Protocol)

It is a Cisco proprietary protocol, so it might not be adequate for an infrastructure where there are other vendors' routers that are not able to support EIGRP.

Although in essence, it's also a distance-vector protocol, it includes some typical link-state features, it's therefore, often classified as a hybrid protocol.

EIGRP path metric's calculation is far more complex than for RIP. For each alternative path, the metric expresses the total path delay and the lowest transmission rate along the path.

Unlike RIP, EIGRP supports autonomous systems identifiers. EIGRP autonomous systems identifiers are numbers between one and 65535. The goal is confining the scope of a dynamic routing protocol to a part of the infrastructure. **Routing tables built within one autonomous system are not spread to other autonomous systems**.

A router can be connected to several networks, use EIGRP in all of them, and yet, each can be a different autonomous system. Of course, different EIGRP autonomous systems identifiers must be used in each of those networks.

For instance, if we have a router connected to networks A and B, and we are using EIGRP on both, we can think about two different scenarios:

We want routing tables built by EIGRP to be transferred between networks A and B.

For this purpose, we will configure a single EIGRP autonomous system and add to it both networks. Networks A and B will belong to the same EIGRP autonomous system (same AS ID's).

We don't want routing tables built by EIGRP to be transferred between networks A and B. Networks A and B must, therefore, belong to different autonomous systems. This router becomes now an Autonomous System Boundary Router (ASBR).

To achieve this, we will configure two EIGRP autonomous systems (each with a different identifier) and add network A to one of them, and network B to the other.

When configuring a set of routers attached to the same EIGRP autonomous system, we must ensure the same identifier is used on all of them.

In Autonomous System Boundary Routers (ASBRs), the construction of routing tables is broken because routing protocols information is not forwarded. What happens is that routing tables, built within and autonomous system will not have any information about networks existing outside that autonomous system.

Fundamental Cisco IOS commands for EIGRP configuration

To create, or edit an existing, EIGRP autonomous system configuration, we can use the following command:

```
(config)#router eigrp AS-IDENTIFIER-NUMBER
(config-router)#
```

Remember, this is only useful if we are going to configure this same EIGRP autonomous system (same AS-IDENTIFIER-NUMBER) all other local routers. Also, this configuration will only have a practical effect after we assign networks to it.

Within the EIGRP router configuration level (config-router), we can now enable EIGRP on directly connected networks by running the following command for each network:

```
(config-router)#network NETWORK-ADDRESS [NETWORK-WILDCARD]
```

Each provided NETWORK-ADDRESS is matched with the router's interfaces addresses, every interface with an address belonging to the network will have EIGRP enforced. This means a single command can be used to enforce EIGRP on several interfaces.

If NETWORK-WILCARD is not specified, then the provided NETWORK-ADDRESS is assumed to be classful, and the corresponding mask is used. The NETWORK-WILCARD is the inverted network mask, but you may also use the normal network mask in dot-decimal representation.

Once a router network interface is assigned to the EIGRP configuration, incoming EIGRP routing tables with matching AS-IDENTIFIER-NUMBER will be incorporated, likewise, the current router's routing table will be sent to other neighbour routers available through the interface.

By default, Cisco routers EIGRP configuration assumes we want to aggregate networks to classful, this is called auto-summary. However, this will be a disaster if we have dispersed classless networks belonging to the same classful address space.

Therefore, unless all networks belonging to the same classful network are attached to the same router, we should always disable this feature:

```
(config-router)#no auto-summary
```

3.1. EIGRP practical exercise. Create the Cisco Packet Tracer diagram in Figure 2.

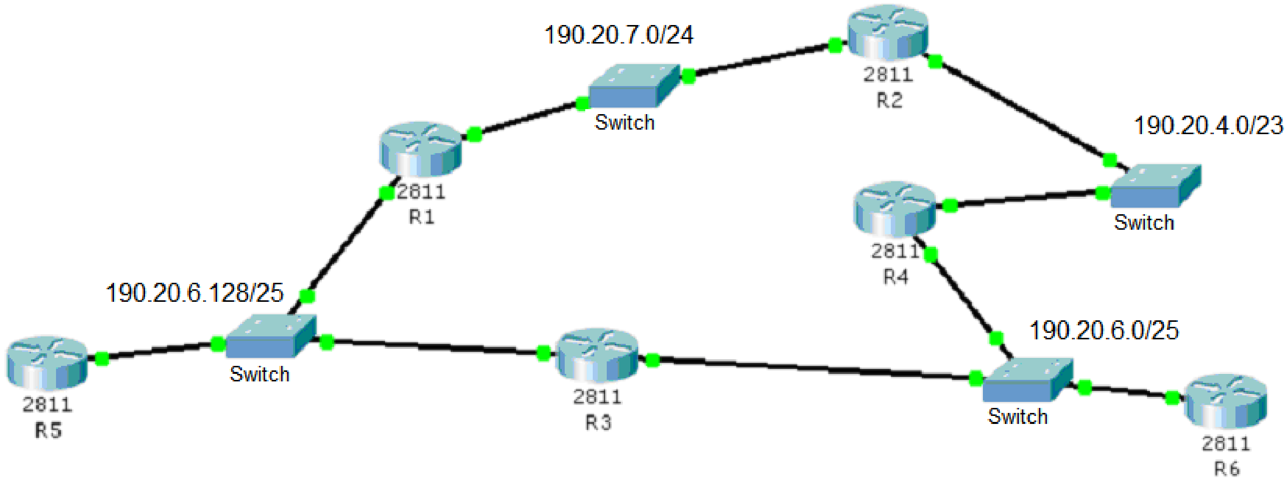

Figure 2 - Networks layout with redundant paths - EIGRP

a) Assign to every router (R1 to R6) valid IPv4 addresses in each connected network.

b) Configure EIGRP autonomous system **200** on all routers and assign all interfaces to it. Notice we are using classless addresses; therefore, **auto-summary should be disabled**.

Use the Inspect tool (magnifying glass) to view each router routing table. All routers routing tables should have four lines except for R5 and R6 that, again, will have 5 lines.

c) In simulation mode send an ICMP echo request from R5 to R6.

Check the path followed by ICMP packets.

d) Shutdown one of R3 network interfaces.

This is going to make it impossible for packets to follow the earlier path.

Now, we must wait for convergence. Use again the Inspect tool to know when routing tables have been changed by RIP.

e) In simulation mode, again, send and ICMP echo request from R5 to R6 to check how packets are traveling now.

f) Now let's go back to the starting point. Enable again the shutdown interface in R3.

Wait for convergence and check that routing tables are now restored to the initial state.

4. OSPF (Open Shortest Path First)

Unlike EIGRP, this is a pure link-state protocol, it's also a public standard. Being a link-state protocol means routing tables are not transmitted from a router to its neighbours. Instead, each router informs other routers about the network around it (neighbour networks and available neighbour routers in each).

Ultimately, this results in every router knowing the whole network infrastructure, at that point each router can build its own routing table.

OSPF also supports autonomous systems. Of course, in relation to other protocols OSPF is always an autonomous system. But one OSPF autonomous system can be divided in areas, each OSPF area is grossly equivalent to an autonomous system.

OSPF areas are identified by 32-bits numbers, sometimes they as represented in dot-decimal as if they were IPv4 addresses. OSPF area zero is reserved for the backbone and must be the first area to be configured, other OSPF areas must be neighbours of area zero.

Fundamental Cisco IOS commands for OSPF configuration

Configuring OSPF in a Cisco router is somewhat similar to EIGRP (and also RIP):

```
(config)#router ospf PROCESS-ID
(config-router)#
```

In this configuration PROCESS-ID is used to identify this specific configuration and makes it possible to have several OSPF configurations running at the same time on the same router.

Nevertheless, PROCESS-ID works as an internal identifier, it has nothing to do with autonomous systems or OSPF areas. The configurations we are going to create are rather simple and a single OSPF configuration per router is enough. You can use PROCESS-ID value one, or any other value, but it may be different in different routers of the same area.

The same way we have done with EIGRP, we must then assign local networks to our OSPF configuration:

```
(config-router)#network NETWORK-ADDRESS NETWORK-WILDCARD area AREA-NUMBER
```

As you can see, the NETWORK-WILDCARD is now required, and the area number is defined for each assigned network.

4.1. OSPF practical exercise.

a) Reuse the diagram from EIGRP practical exercise. Disable EIGRP on all routers (R1 to R6).

```
(config)#no router eigrp 200
```

b) Configure OSPF on all routers and assign all interfaces to it, on area zero.

Use the Inspect tool (magnifying glass) to view each router routing table. All routers routing tables should have four lines except for R5 and R6 that, again, will have 5 lines.

c) In simulation mode send an ICMP echo request from R5 to R6.

Check the path followed by ICMP packets.

d) Shutdown one of R3 network interfaces.

This is going to make it impossible for packets to follow the previous path.

Now, we must wait for convergence. Use again the Inspect tool to know when routing tables have been changed by RIP.

e) In simulation mode, again, send and ICMP echo request from R5 to R6 to check how packets are traveling now.

f) Now let's go back to the starting point. Enable again the shutdown interface on R3.

Wait for convergence and check that routing tables are now restored to the initial state.

5. Autonomous Systems

We already have a fair idea why autonomous systems are required. Routing protocols are extremely useful in building and keeping routing tables updated, and thus, providing fault tolerance and even network load balancing. Yet, the number of networks managed by a routing protocol deployment has to be restrained.

As the number of networks increases, some performance issues will arise:

- Routing tables get bigger.
- Convergence time gets longer.
- Routing protocol traffic increases.

To avoid these problems, limits to routing protocols propagation are settled. These limits define an autonomous system.

An autonomous system is an infrastructure zone, made of several interconnected networks, in which a routing protocol is deployed to dynamically manage routing tables. We can also say an autonomous system is a set of adjacent networks, networks outside the autonomous system will not be included in its routing tables.

5.1. Autonomous System Boundary Router (ASBR)

Autonomous systems limits are enforced by routers. Routers with this role are called Autonomous System Boundary Routers. An ASBR is a normal router, however, its network interfaces are not all assigned to the same autonomous system.

A router may interact with different autonomous systems, either because, it uses different routing protocols (e.g., RIP and EIGRP), or because it's using a same routing protocol, but with different autonomous system identifiers.

RIP (Routing Information Protocol) lacks any autonomous system identification; this doesn't mean it can't be used to create an autonomous system. The only constrain around RIP based autonomous systems is that they cannot be adjacent. That's so because it's not possible to have two RIP autonomous systems connected to the same router, they would simply become the same.

5.2. Autonomous System Number (ASN)

Several routing protocols identify autonomous systems by numbers, known as autonomous system numbers.
Currently, Border Gateway Protocol (BGP) is the standard exterior gateway protocol over the internet. BGP border routers connect local autonomous systems to the internet, each local autonomous system has an IANA assigned ASN. Originally BGP ASNs where 16-bits numbers, but they have now been upgraded to 32-bits numbers.
EIGRP also needs a 16-bits ASN, although not related to BGP ASNs, they serve the same purpose of identifying an autonomous system.
EIGRP ASN numbers can be from one up to 65535. They allow an autonomous system boundary router to be connected to several different EIGRP autonomous systems, as far as each uses a different ASN number.

5.3. OSPF area numbers

OSPF is somewhat different, but after all, equivalent. A set of adjacent networks being managed by OSPF is an **OSPF Domain**. Under other routing protocols perspective, one OSPF Domain is one autonomous system.

We should emphasise **adjacent networks**, otherwise it will no longer be a single OSPF Domain. This issue can though be overcome, by using virtual links between routers (an advanced feature of OSPF configuration, not being detailed here).

At first glance, it looks pretty the same as RIP. Nevertheless, an OSPF Domain can be divided into OSPF areas.

Some autonomous systems goals are shared with OSPF areas, namely, reduce routing tables' length, routing protocol traffic and the convergence time. Like with an autonomous system, an OSPF area should also be made of a single IP addresses block. The main difference is that, unlike what happens between autonomous systems, **routing information is automatically forwarded between OSPF areas**.

Each OSPF area is identified by a 32-bits number, they are often presented in dot-decimal notation. The first area to be created must be area zero (0.0.0.0), it's known as **backbone area**. Other areas must be physically adjacent (or virtual linked) to the backbone area.

Area Border Routers (ABR) are routers with one interface attached to the backbone area and other interfaces attached to other OSPF areas, as already mentioned, they ensure automated routing information forwarding between areas.

6. Practical exercise

Create the Cisco Packet Tracer diagram shown in Figure 3, with routers and layer two devices.

(This diagram is available for download, with already settled IP addresses: pl06-a.pkt)

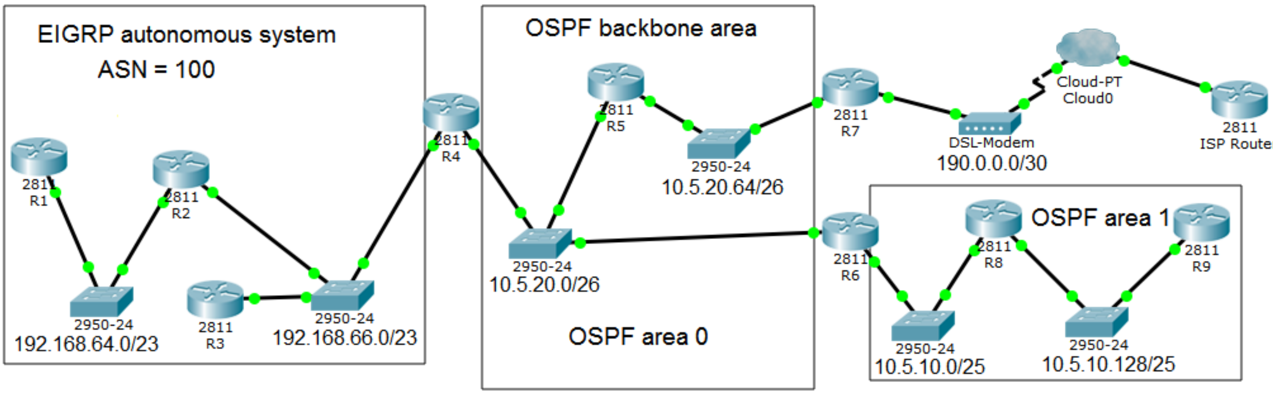

Figure 3 - Autonomous systems

IPv4 network addresses are presented at the image, below layer two devices.

6.1. Assign a valid IPv4 node address to every router's interface (only if starting from scratch).

Concerning the ISP router connection, assign to it address 190.0.0.2/30. Assign router R2 in 192.168.66.0/23 network, the 192.168.66.1 address. Others assign at your will, as far as they are valid non-overlapping node addresses.

6.2. Setup the EIGRP autonomous system presented at the image.

Router R4 is going to play an ASBR (Autonomous System Boundary Router) role, therefore, only one interface belongs to the autonomous system.

R1 (config)#router eigrp 100 (config-router)#no auto-summary (config-router)#network 192.168.64.0 0.0.1.255	R2 (config)#router eigrp 100 (config-router)#no auto-summary (config-router)#network 192.168.64.0 0.0.1.255 (config-router)#network 192.168.66.0 0.0.1.255
R3 (config)#router eigrp 100 (config-router)#no auto-summary (config-router)#network 192.168.66.0 0.0.1.255	R4 (config)#router eigrp 100 (config-router)#no auto-summary (config-router)#network 192.168.66.0 0.0.1.255

6.3. Setup OSPF areas 0 and 1 as presented at the image.

Again, router R4 is going to play an ASBR (Autonomous System Boundary Router) role, therefore, only one interface belongs to OSPF area 0.

Router R6 will be an area border router (ABR), one interface will be on area 0 and the other in area 1.

Router R7 will also be an ASBR, although it will be connected to a single autonomous system.

R4	R5
(config)#router ospf 5 (config-router)#network 10.5.20.0 0.0.0.63 area 0	(config)#router ospf 5 (config-router)#network 10.5.20.0 0.0.0.63 area 0 (config-router)#network 10.5.20.64 0.0.0.63 area 0
R6	R7
(config)#router ospf 5 (config-router)network 10.5.20.0 0.0.0.63 area 0 (config-router)#network 10.5.10.0 0.0.0.127 area 1	(config)#router ospf 5 (config-router)network 10.5.20.64 0.0.0.63 area 0
R8	R9
(config)#router ospf 5 (config-router)network 10.5.10.0 0.0.0.127 area 1 (config-router)network 10.5.10.128 0.0.0.127 area 1	(config)#router ospf 5 (config-router)network 10.5.10.128 0.0.0.127 area 1

6.4. Check each router's routing table.

Use the Inspect tool on each router (or the **show ip route** command at CLI).

Draw your conclusions.

We can see within the OSPF domain, routing tables are ok, all networks are reachable. OSPF ensures routing information forwarding between areas.

Within the EIGRP autonomous system, routing tables are ok, as well.

Though there are some issues:

- In the OSPF domain, networks belonging to the EIGRP autonomous system are unknown.
- In the EIGRP autonomous system, networks belonging to the OSPF domain are unknown.
- A default route for the internet connection is missing.

7. Route redistribution

Autonomous System Boundary Routers (in our example, R4 and R7) block routing information propagation (as they are supposed to).

Nevertheless, ultimately, routing has to work between networks, even if they belong to different autonomous systems. We achieve this by inserting routing information into the autonomous system routing protocol, usually at boundary routers. This is called **route redistribution**.

In Cisco IOS we can enforce **routes redistribution** by using the **redistribute** command, within the routing protocol configuration level, in which we want to insert rules:

(config-router) **redistribute** {protocol-to-redistribute}

Where {protocol-to-redistribute} represents the routing protocol (autonomous system) from which we want automatic copy of routing information to the present autonomous system.

By default, the redistribute command inserts only classful network addresses, to handle classless addresses, the **subnets** parameter must be added.

We can also insert static routing information by using the **redistribute static** command. Local networks can also be inserted by using the **redistribute connected** command. For default route insertion, on the autonomous system boundary router that has the default route defined we can use the **default-information originate** command to insert it:

(config-router) **default-information originate**

One issue we must overcome is that inserted information must have the right metric for the routing protocol we are inserting into. Unless we are copying routing information between autonomous systems using the same protocol, the metric value will have to be manually assigned.

8. Practical exercise - adding route redistribution

8.1. Define the default route and redistribute it

Router R7 is the autonomous system boundary router that provides internet connection to the ISP router, therefore, it's the most suitable place to insert the default route into OSPF area 0.

```
                              R7

(config)#ip route 0.0.0.0 0.0.0.0 190.0.0.2
(config)#router ospf 5
(config-router)# default-information originate
```

Check that, now, every OSPF domain router has a default route in the routing table.

8.2. Redistribute OSPF domain information into the EIGRP autonomous system

This can be achieved at router R4 because it's the autonomous system boundary router that connects the EIGRP autonomous system to the OSPF autonomous system.

```
                                                R4

(config)#router eigrp 100
(config-router)# redistribute ospf 5 metric 100000 100 255 1 1500
```

EIGRP metric is not available at OSPF domain, so we must define it manually using the metric parameter. The sample values above mean:

100000 (transmission rate in kbps) – 100 Mbps
100 (network delay in 10 microseconds units) – 1 millisecond
255 (reliability from 0 up to 255) – 100%
1 (network load from 1 up to 255) – 0 %
1500 (MTU value in bytes) – 1500 bytes

Check that, now, routers in the EIGRP autonomous systems know about networks inside the OSPF domain, including the default route inserted by R7 into the OSPF domain.

However, the OSPF domain is still not aware of the EIGRP autonomous system networks. One more step is still required.

8.3. Redistribute static information into the OSPF domain

We could do as before for inserting EIGRP routing information into the OSPF domain. We would enter into the OSPF configuration in router R4 and **redistribute eigrp 100**.

But we will use another approach instead: static information routes insertion:

```
                                                R4

(config)#ip route 192.168.64.0 255.255.254.0 192.168.66.1
(config)#router ospf 5
(config-router)# redistribute static subnets
(config-router)# redistribute connected subnets
```

Now, networks 192.168.64.0/23 and 192.168.66.0/23 are known in the OSPF domain. The first, due to **redistribute static** command, and the second due to **redistribute connected** command.

Apart from the ISP router (outside our scope), every node is now able to communicate with any other node. Also, every packet sent to a destination address not belonging to any under scope network will be delivered to the ISP router, as it's supposed to be.

9. Cisco IOS Telephony Services (ITS)

VoIP configuration is rather vendor specific, moreover, within the same vendor different device models may require different configurations. We are using the Cisco 2811 router and the Cisco 7960 IP phone.

Cisco phones use the Skinny Client Control Protocol (SCCP) to communicate with the Cisco CallManager Express (CME) service, this service is going to be running on the 2811 router. By default, the service is provided at port number 2000.

9.1. Switches configuration

Cisco VoIP devices like the **7960 model** available in Packet Tracer require VoIP packets to be encapsulated in specially formatted Ethernet frames (with Layer 2 CoS priority value).

In a Cisco switch, every port attached to as IP phone must be configured to use that format, this is achieved by enabling the **voice vlan** on that port and disabling the **access vlan**. Also, **the port should be in access-mode, not trunk-mode**.

Example: assume the VoIP VLAN is **VLANID=300** and is available at switch in Figure 4.

Then, **Switch3 Fa0/2** port must in **access-mode**, have the **voice vlan** enabled (it's disabled by default). Fa0/2 must also have the **access vlan** disabled (it's enabled by default).

This setup may be achieved with the following commands:

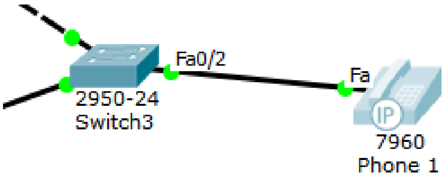

Figure 4 - 7960 VoIP phone connected to a 2950 switch

```
Switch(config)#interface Fa0/2
Switch(config-if)#
Switch(config-if)#switchport mode access
Switch(config-if)#switchport voice vlan 300
Switch(config-if)#no switchport access vlan
```

However, this port configuration is required only on switch's ports directly connected to VoIP phones, not on all switch ports that use the VoIP VLAN.

9.2. DHCP service configuration

Any device can provide the DHCP service to VoIP phones, but an additional option must be added to provide the client with information about the TFTP (Trivial File Transfer Protocol) server to be used by the phone. The IST (IOS Telephony Services) server also provides the TFTP service. In Packet Tracer, the 2811 model router can function as IST server, it will also provide the DHCP service, and hence, it must be directly connected to the VoIP VLAN where phones are connected.

The DHCP option 150 holds an IPv4 address of the TFTP server from where clients are supposed to download configuration data.

The DHCP option 150 is Cisco proprietary, the equivalent IEEE standard is DHCP option 66. Though, option 66 allows a single address, for a single TFTP server, whereas option 150 allows a list of addresses, for a list of alternative TFTP servers. Anyway, for Cisco VoIP devices, option 150 must be used.

Example: If the 2811 router is connected to a VoIP VLAN with IPv4 address 10.40.50.0/24 and it's using address 10.40.50.1, then it can provide the DHCP service, the TFTP service and the router could be also the network's default-gateway. In this case the router will be the IST server as well.

The configuration might look like:

```
Router(config)#ip dhcp pool MYVOIP
Router(dhcp-config)#default-router 10.40.50.1
Router(dhcp-config)# option 150 ip 10.40.50.1
Router(dhcp-config)# network 10.40.50.0 255.255.255.0
```

9.3. ITS service configuration

Cisco Telephony Services configuration may differ depending on the models of the IP phone and router used.

By default, new VoIP phones are automatically registered by the service, this is controlled by the **auto-reg-ephone** command. Once registered, the phone MAC address is stored, and that phone will always be the same. If automatic registration is not used, then the MAC address for each phone must be manually set.

The Cisco Telephony Services configuration requires the declaration of the IST server IPv4 address and port number for phones to register, this is settled by the **ip source-address** command. The default service port number is 2000.

Other important settings in Telephony Services configuration is the maximum number of supported phones (**max-ephones**), and the maximum number of supported directory numbers (**max-dn**), meaning the maximum number of phone numbers.

More directory numbers (phone numbers) than phones may be needed because a single phone can have several lines (referred to as buttons), each line with a different directory number (though this is not supported in Packet Tracer). Of course, directory numbers must be unique, and each can be assigned to only one phone line (one phone's button).

If automatic phone registration is used, then automatic assignment of directory numbers can also be enforced by declaring a range of directory numbers for that purpose (**auto assign** command).

Phones are identified by numbers from one up to **max-ephones**, also directory numbers are identified by numbers from one up to **max-dn**. However, there is no direct relation between a phone id number and a directory number id. Directory numbers must be assigned to phones (phone lines).

Once **max-dn** and **max-ephones** have been settled, phones and directory numbers can be declared.

A directory number is declared by using the **ephone-dn** command and then the **number** command to assign a number to be dialled. Example for configuring directory number with id 11 to be 945073:

```
Router(config)#ephone-dn 11
Router(config-ephone-dn)number 945073
```

A phone is declared by using the **ephone** command, then the device model can be specified, the device mac address and a directory number can be assigned to the phone. Example:

```
Router(config)#ephone 5
Router(config-ephone)type 7960
Router(config-ephone)mac-address 00D0.976D.1BC1
Router(config-ephone)button 1:11
```

Together with the previous directory number declaration, this means when a 7960 model with MAC address 00:D0:97:6D:1B:C1 is connected to the network it will be registered as phone with id 5 and the phone number assigned to it (line 1/button 1) is going to be directory number with id 11, so it will have the phone number 945073.

There are a few different methods for Cisco IP phones configuration and corresponding directory numbers assignment.

- **Totally manual**: set up each phone's MAC address and directory number.
- **Automatic registration**: phones are added automatically, and directory numbers assigned later, manually.
- **Automatic registration and automatic directory numbers assignment**.

Option A - Manual phone configuration and number assignment

If automatic registration is disabled, then each phone MAC address must be manually set (**mac-address** command), and directory numbers must also be manually assigned (button command).

Example:

```
Router(config)#telephony-service
Router(config-telephony)#no auto-reg-ephone
Router(config-telephony)#ip source-address 100.100.100.1 port 2000
Router(config-telephony)#max-ephones 20
Router(config-telephony)#max-dn 20

Router(config)#ephone-dn 11
Router(config-ephone-dn)number 945073

Router(config)#ephone-dn 12
Router(config-ephone-dn)number 945074

Router(config)#ephone 5
Router(config-ephone)mac-address 00D0.976D.1BC1
Router(config-ephone)button 1:11

Router(config)#ephone 3
Router(config-ephone)mac-address 00D0.976D.ABC6
Router(config-ephone)button 1:12
```

This results in, phone with MAC address 00:D0:97:6D:1B:C1 will be using number 945073, and phone with MAC address 00:D0:97:6D:AB:C6 is going to have dial number 945074.

Option B - Automatic phone registration and manual directory number assignment

With automatic registration enabled, phones are declared, but not theirs' MAC addresses. The directory number of each phone can only be assigned after registration.

```
Router(config)#telephony-service
Router(config-telephony)#auto-reg-ephone
Router(config-telephony)#ip source-address 100.100.100.1 port 2000
Router(config-telephony)#max-ephones 20
Router(config-telephony)#max-dn 20

Router(config)#ephone-dn 11
Router(config-ephone-dn)number 945073

Router(config)#ephone-dn 12
Router(config-ephone-dn)number 945074

Router(config)#ephone 5
Router(config-ephone)

Router(config)#ephone 3
Router(config-ephone)
```

With this configuration, new phones will be spontaneously assigned to empty **ephone** declarations, and the **mac-address** command is automatically added to each declaration. Because these assignments are added to the router's configuration, they become permanent.

By not including a range of directory numbers for automatic assignment, registered phones will be offline until a directory number is manually assigned. That can only be done after the phone registers.

When new phones are registered, they will show up by using the **show ephone** command in **privileged EXEC mode**. Also, by checking the running configuration (show running-config command) the corresponding **ephone** declaration will have a MAC address defined.

After a new phone being registered, for instance as ephone 5, then a directory number can be manually assigned to it, say directory number with id 11:

```
Router(config)#ephone 5
Router(config-ephone)button 1:11
```

Option C - Automatic phone registration and directory number assignment

With automatic registration enabled, ranges of directory numbers to be automatically assigned to newly registered phones must be declared. In the following example, directory numbers 11 up to 12 will be used:

```
Router(config)#telephony-service
Router(config-telephony)#auto-reg-ephone
Router(config-telephony)#ip source-address 100.100.100.1 port 2000
Router(config-telephony)#max-ephones 20
Router(config-telephony)#max-dn 20
Router(config-telephony)#auto assign 11 to 12

Router(config)#ephone-dn 11
Router(config-ephone-dn)number 945073

Router(config)#ephone-dn 12
Router(config-ephone-dn)number 945074
```

Now, new phones are registered automatically, and in addition, to each newly registered phone an unused directory number from the provided range will be assigned. The corresponding button command is also added to the ephone declaration, so this assignment becomes permanent as well.

Afterwards, the phone number assigned to each phone may be easily changed, simply by changing the **number** command within the corresponding **ephone-dn** declaration.

9.4. Calls forwarding

Phones registered in different ITS servers, can make calls between each other's. For this to be possible, each ITS server must be aware of other existing ITS servers, and phone numbers used by each.

This is achieved by setting up **addressable call endpoints** or **dial peers** on servers running the ITS services, the **dial-peer** command is used to declare a dial peer. Each dial peer is identified by tag, it's an internal number (between 1 and 2147483647) with no special meaning beyond identifying one specific dial peer.

To forward VoIP incoming calls to other ITS servers the dial-peer command to be used is:

```
dial-peer voice TAG voip
```

This will enter in that (TAG) specific dial peer configuration mode, then, for our purpose, two things must be established:

- Which dial numbers will match this dial peer, this is achieved through the **destination-pattern** command, and it may either specify a single fixed phone number or a pattern. A pattern is made of fixed digits combined with dots and other pattern matching characters. A dot matches any digit, but not an empty digit. For instance, pattern 4… matches 4444, but not 444.

- A call forwarding target, meaning to which ITS server should be matching incoming calls forward to. This is achieved through the **session target ipv4:IP-address** command, where IP-address is the IPv4 address of the ITS server to which the calls are to be forwarded to.

Within each dial-peer declaration only one destination-pattern and one session target can be declared.

Example: If we want the local ITS server to forward incoming calls to any phone number with 5 digits starting with digits 53, to be forwarded to ITS server 192.168.10.5.

Then the following commands would settle an appropriate dial peer:

```
dial-peer voice 10 voip
    destination-pattern 53…
    session target ipv4:192.168.10.5
```

Dial peers are used in both directions, for inbound calls and outbound calls and they must match.

The above configuration establishes, not only, that local calls with destination phone numbers 53… are to be forwarded to ITS server 192.168.10.5, but also that calls will be received from origin phone numbers 53…, this means that for the above configuration to work, in ITS server 192.168.10.5 there must be also a dial peer pointing out to our local server and matching our local phone numbers.

9.5. Packet Tracer practice

Let's now put all these concepts into practice, download from Moodle (pl06-b.pkt) the Packet Tracer layout in Figure 5. Routers RT1 and RT2 have IPv4 addresses already settled as show in the image, but no additional configurations have been enforced. Everywhere, only the default VLAN is being used (VLANID=1).

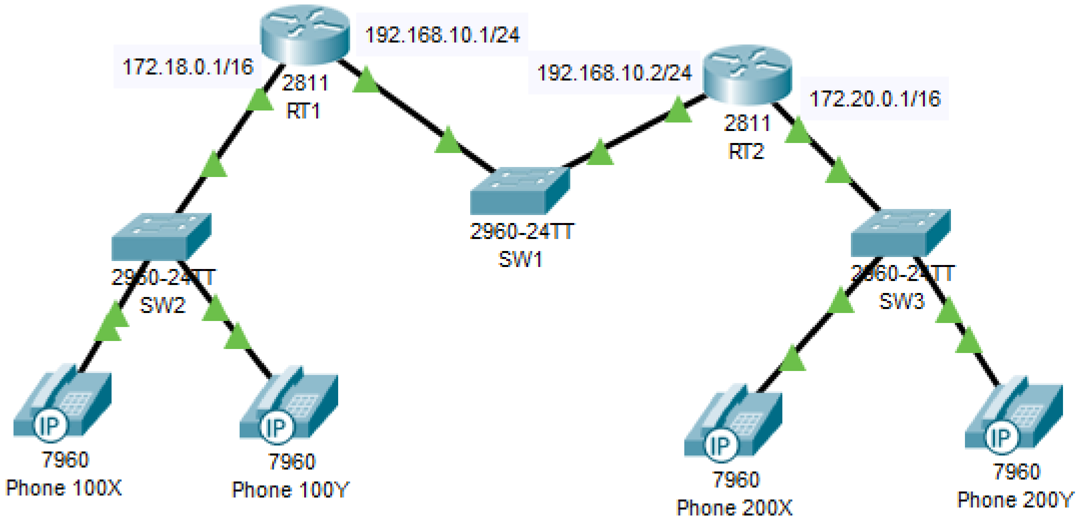

Figure 5 - VoIP layout

Our aim is running the ITS service on router RT1, to serve the left side network IP phones, and also run the ITS service on router RT2, to serve the right-side network IP phones.

The left side IP phones are going to use phone numbers 1000 and 1001, the right-side phones are going to use phone numbers 2000 and 2001, yet because we are going to use automatic dial numbers assignment, we don't know which phone is going to get each number.

Finally, we will set up dial peers to forward calls between phones served by the two different ITS servers.

a) **Configure SW2 and SW3 ports that are connected to IP phones (Fa0/2 and Fa0/3) as explained before:**

```
int fa0/2
      switchport mode access
      switchport voice vlan 1
      no switchport access vlan
int fa0/3
      switchport mode access
      switchport voice vlan 1
      no switchport access vlan
```

(This has to be enforced on both switches, by chance, in this case the interfaces are the same on both switches)

b) **Create DHCP pools on both routers to serve local networks, for instance:**

RT1:

```
ip dhcp excluded-address 172.18.0.1 172.18.0.255
ip dhcp pool VOIP
       option 150 ip 172.18.0.1
       network 172.18.0.0 255.255.0.0
```

RT2:

```
ip dhcp excluded-address 172.20.0.1 172.20.0.255
ip dhcp pool VOIP
       option 150 ip 172.20.0.1
       network 172.20.0.0 255.255.0.0
```

Notice that no routing configurations have been enforced because it is not really necessary in this specific case, IP phones on different networks are not going to talk to each other's directly, they will use each local ITS server to forward phone calls.

c) **Use the "Automatic phone registration and directory number assignment" method to register phones and assign dial numbers, both on RT1 and RT2 ITS services.**

RT1:

```
telephony-service
       max-ephones 2
       max-dn 2
       ip source-address 172.18.0.1 port 2000
       auto assign 1 to 2

ephone-dn 1
       number 1000

ephone-dn 2
       number 1001
```

RT2:

```
telephony-service
       max-ephones 2
       max-dn 2
       ip source-address 172.20.0.1 port 2000
       auto assign 1 to 2

ephone-dn 1
       number 2000

ephone-dn 2
       number 2001
```

If everything went as expected, all four phones should be registered and have dial numbers assigned. Flyby the mouse over IP phones to see if they have an IPv4 address, and a phone number assigned.

You may also run the **show ephone** command on routers to see the current status.

Now, test local phone calls:

- From phone number 1000 to 1001 and vice versa (on the left side network).
- From phone number 2000 to 2001 and vice versa (on the right-side network).

They should work.

However, phone calls to remote phone numbers will not work for now, e.g., from phone number 1000 to 2000.

d) Configure phone calls forwarding between the two ITS servers.

For instance:

RT1:

```
dial-peer voice 20 voip
       destination-pattern 2...
       session target ipv4:192.168.10.2
```

RT2:

```
dial-peer voice 1 voip
       destination-pattern 1...
       session target ipv4:192.168.10.1
```

Now, test remote phone calls, e.g.:

- From phone number 1000 to 2001 and vice versa.

Now they should work.