CHAPTER 9

CONFIGURING CISCO SWITCHES

9 Configuring Cisco Switches

The chapter covers basic theoretical and practical knowledge of configuring simulated switches in the Cisco Packet Tracer.

Note – Warning: Remember to periodically save the file state during exercises (keyboard shortcut CTRL+S)

9.1 Exploring the Equipment of Cisco Switches

Another type of devices available in the program are **switches**. We have at our disposal three basic switches that are equivalent to real models from the **2950** to **2960** series, models called **PT-Switch** and **PT-Empty**, which can be modified in any way, as well as third layer switches **3560**, **3650** and **IE 2000**.

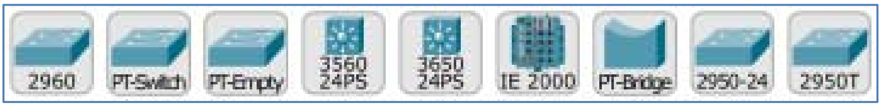


Figure 9.1 Switch models available in the Cisco Packet Tracer

Unlike routers, each of which had the ability to install optional modules, in the case of switches, only **PT** models allow the replacement of modules. Other devices do not allow you to change their equipment.

9.1.1 Switch 2950

The first switch we will discuss is the 2950 series switch. It is a manageable device dedicated to small and medium-sized networks. It has 24 Fast Ethernet interfaces with a bandwidth of 10/100 Mbps.

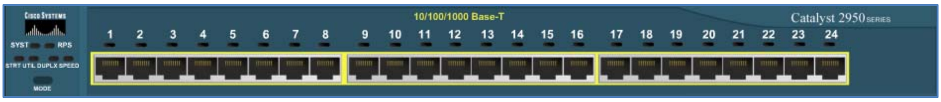


Figure 9.2 Physical appearance of the 2950 series switch

The 2950 Series Switch does not support any optional equipment.

9.1.2 Switch 2950T

Another switch available in the program is the 2950T series switch. This switch is a managed device dedicated to a medium-sized network. It has 24 Fast Ethernet interfaces with a bandwidth of 10/100 Mbps, as well as two Gigabit Ethernet interfaces with a bandwidth of 10/100/1000 Mbps.

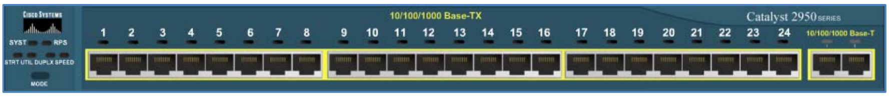


Figure 9.3 Physical appearance of the 2950T series switch

The 2950T Series Switch does not support any optional equipment.

9.1.3 Switch 2960

Another switch available in the program is the 2960 series model. It is a manageable device dedicated to medium-sized networks. It has 24 Fast Ethernet interfaces with a bandwidth of 10/100 Mbps, as well as two Gigabit Ethernet interfaces with a bandwidth of 10/100/1000 Mbps similar to the switch version 2950.

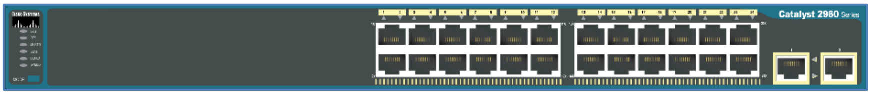


Figure 9.4 Physical appearance of the 2960 series switch

The 2960 Series Switch does not support any optional equipment.

9.1.4 PT-Switch and PT-Empty Switch

The next switches available in the program are the two PT series switches, which are devices only found in the PT program, but allow you to create many different hardware configurations. They have ten slots, in the place of which we can mount the selected interface. When adding this switch to the topology, there are two versions to choose from: **PT-Switch** and **PT-Empty**. These switches also have power switches.

The first of them by default has four Fast Ethernet ports with a bandwidth of 10/100 Mbps, using twisted pair cables and two Fast Ethernet interfaces with a bandwidth of 10/100 Mbps in the fiber optic standard and two free slots. The second is a completely empty version, without mounted modules.

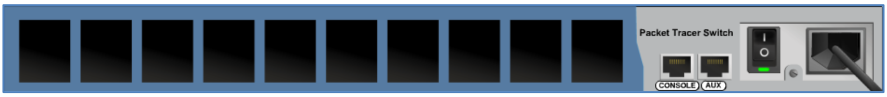
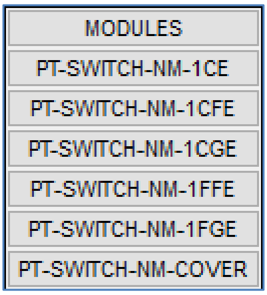


Figure 9.5 Physical appearance of the PT-Empty switch

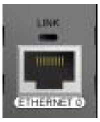
In free slots we can install the following components:

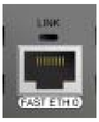


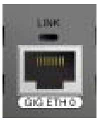
MODULES
PT-SWITCH-NM-1CE
PT-SWITCH-NM-1CFE
PT-SWITCH-NM-1CGE
PT-SWITCH-NM-1FFE
PT-SWITCH-NM-1FGE
PT-SWITCH-NM-COVER

Figure 9.6 PT Switch Optional Equipment

Below is a detailed description of the individual components:

- 

PT-SWITCH-NM-1CE - a single Ethernet port with a bandwidth of 10 Mbps,
- 

PT-SWITCH-NM-1CFE - single Fast Ethernet port with a bandwidth of 10/100 Mbps,
- 

PT-SWITCH-NM-1CGE - single Gigabit Ethernet port with 10/100/1000 Mbps bandwidth,
-

PT-SWITCH-NM-1FFE - single Fast Ethernet port with a bandwidth of 10/100 Mbps in the fiber optic standard,

- **PT-SWITCH-NM-1FGE** - a single 10/100/1000 Mbps Gigabit Ethernet port in fiber optic standard.
- **PT-SWITCH-NM-COVER -** module for protecting the inside of the switch

9.1.5 Series 3560

Another switch available in the program is the 3560 series switch. It is a so-called multi-layer switch, which means that it has wider possibilities compared to other switches, because it works not only in the second ISO / OSI layer, but also in the third layer (in the same as routers). It has 24 Fast Ethernet interfaces with a bandwidth of 10/100 Mbps, as well as two Gigabit Ethernet interfaces with a bandwidth of 10/100/1000 Mbps.

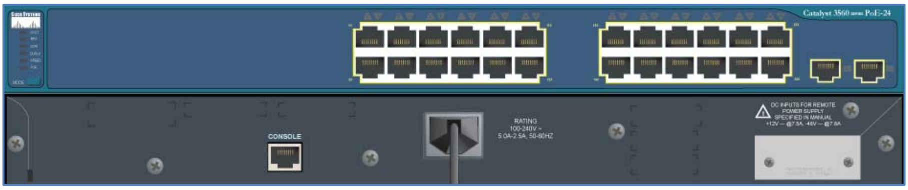


Figure 9.7 Physical appearance of the 3560 series switch

The 3560 Series Switch does not support any optional equipment.

9.1.6 Series IE 2000

The **IE 2000** (*Industrial Ethernet 2000 Series*) switch is an industrial switch that simulates a 2000 series switch model. The switch is running IOS **IE2000 Software (IE2000-UNIVERSALK9-M), version 15.2(1)EY**. The switch meets the requirements of industrial networks for speed of convergence and security.

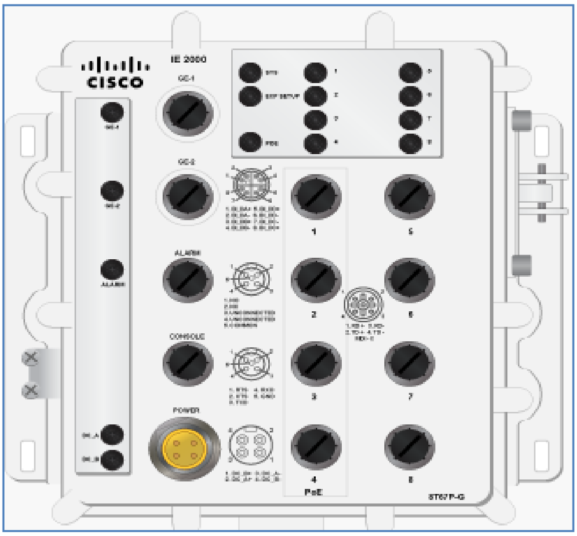


Figure 9.8 Physical appearance of the IE 2000 series switch

The switch has 8 Fast Ethernet interfaces and 2 Gigabit Ehternet interfaces. Supports Resilient Ethernet Protocol (**REP**).

9.2 Configure Cisco Switches Using the Graphical Interface

9.2.1 Interface Configuration

At the beginning we will create a simple topology consisting of four computers and one switch.

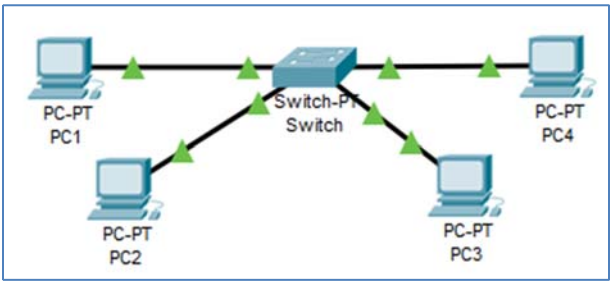
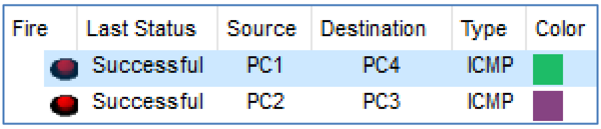


Figure 9.9 Topology for switch configuration

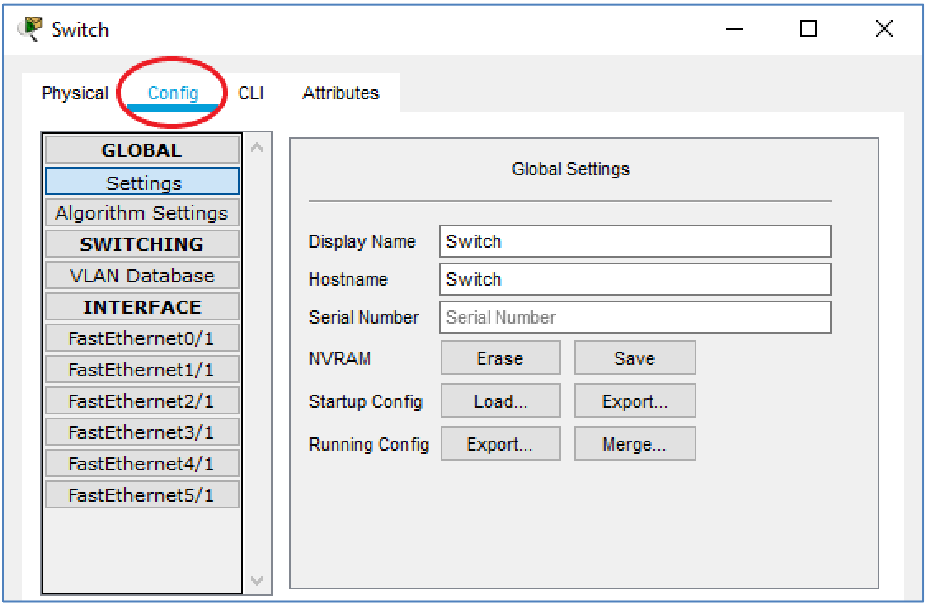
As we already know, when we address these computers now (for example, so that they are in the network 192.168.0.0/24), they will be able to communicate with each other without any problem as shown in the figure below – despite the fact that the configuration of the switch remained the default.



Fire	Last Status	Source	Destination	Type	Color
●	Successful	PC1	PC4	ICMP	🟩
●	Successful	PC2	PC3	ICMP	🟪

Figure 9.10 PING between devices

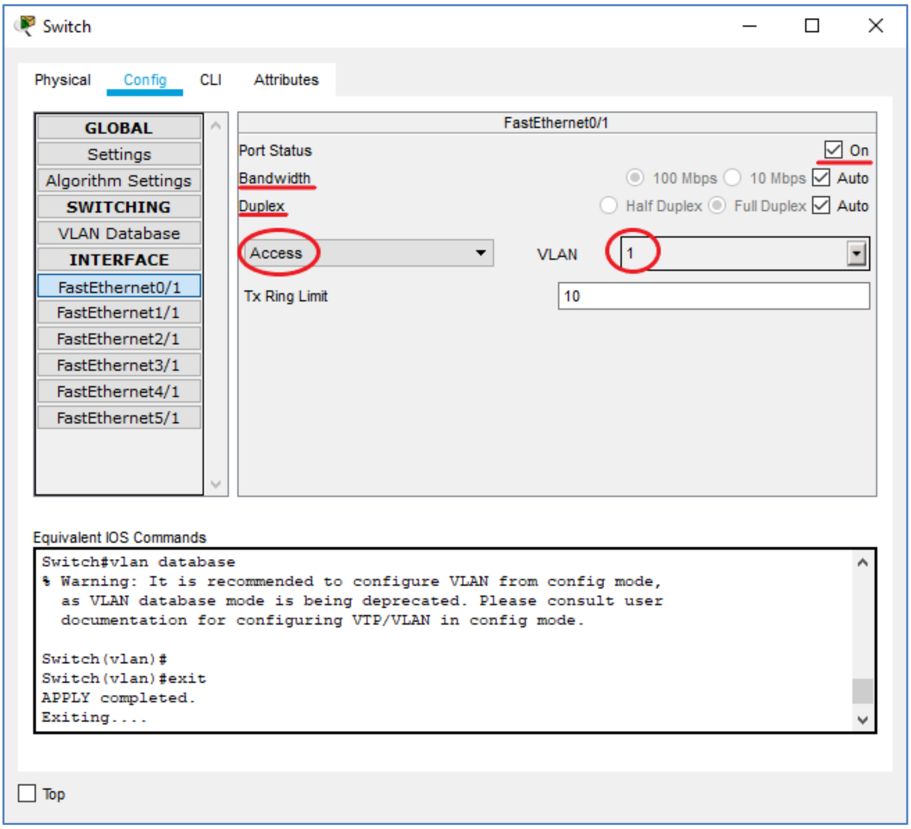
Now that we are sure that with the default configuration our network works seamlessly, we can go to the **Config** tab on the switch and see what options the configuration offers us using the GUI.



Display Name: Switch
Hostname: Switch
Serial Number: Serial Number
NVRAM
Startup Config
Running Config

Figure 9.11 Switch Config tab

As you can see, we do not have too many options to choose from - configuration of basic settings such as the name of the switch or saving the current configuration, then there is **Algorithm Settings** - not the option we are interested in, **VLAN database** and configuration of the interfaces themselves. Let's take a look at the interface configuration at the beginning.



FastEthernet0/1

Port Status ☑ On

Bandwidth ● 100 Mbps ○ 10 Mbps ☑ Auto

Duplex ○ Half Duplex ● Full Duplex ☑ Auto

Access VLAN 1

Tx Ring Limit 10

Equivalent IOS Commands

```
Switch#vlan database
% Warning: It is recommended to configure VLAN from config mode,
  as VLAN database mode is being deprecated. Please consult user
  documentation for configuring VTP/VLAN in config mode.

Switch(vlan)#
Switch(vlan)#exit
APPLY completed.
Exiting....
```

Figure 9.12 Switch Interface Configuration - GUI

We have at our disposal enabling / disabling the interface, with what bandwidth a given port works, what duplex it supports and what VLAN belongs to and in what mode the port is located (we have a choice of **Access** or **Trunk** mode, as in the figure below).

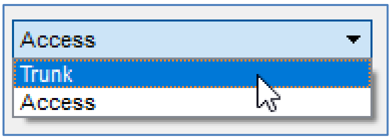


Figure 9.13 Selecting the port mode

The difference in the operation of these modes is described below:

- **Access -** if the port is in this mode, the switch usually accepts all untagged frames from this port and gives them a predefined tag - depends on which VLAN the port is located in. If data is to be sent to a port that is in this mode, the tag is removed. Most often, access mode is used on ports to which end users are attached, e.g. desktops, laptops, printers.

- **Trunk -** a port through which information from all VLANs can be sent by default. In this port pass frames that are tamed and usually ports of this type connect two switches, or a switch to the router.

Tagging frames involves adding an additional field (VLAN tag) containing the VLAN number to them. In this example, the ports work in the default Access mode.

9.2.2 Configuring Virtual LANs (VLANs)

What is a **VLAN** (Virtual LAN) and how do I configure it? This is one of the most useful functions of switches that we can configure, so it is worth getting to know this issue and mastering it as much as possible.

As the name suggests, these are LANs, but separated from each other virtually, not physically. For example – in a residential building we have many LANs, as a standard each apartment is a separate LAN, and in the case of VLANs we can virtually divide one apartment into smaller LANs that cannot communicate with each other without the participation of a router.

To illustrate the division of LANs into VLANs, we will use the same topology that was presented a moment ago, but we will divide it in the diagram into two separate VLANs: **VLAN10** and **VLAN20**.

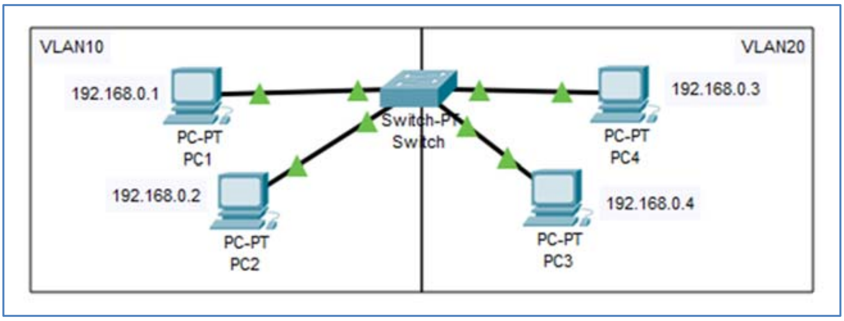
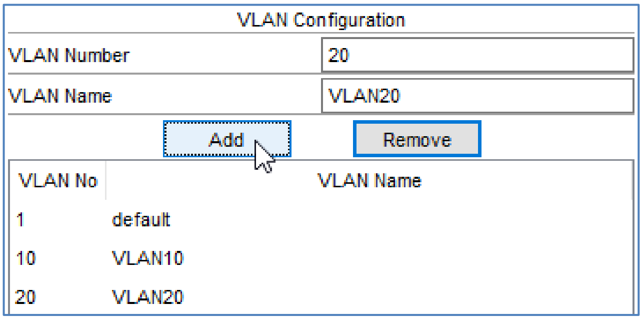


Figure 9.14 Topology for VLAN configuration (Example 9.2.2a.pkt)

Now you need to configure VLANs on the switch. In the **Config** tab, go to the **VLAN Database**. There we need to add two VLANs that interest us: number **10** and number **20** with the names **VLAN10** and **VLAN20**. respectively.

VLAN number 1 is the default VLAN and includes all ports before the switch is configured.



VLAN Configuration

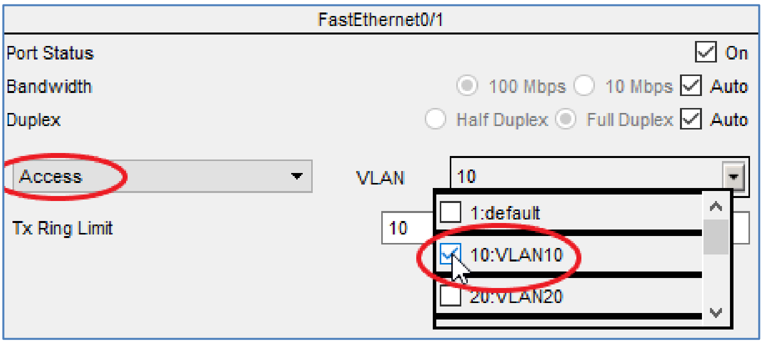
VLAN Number: 20

VLAN Name: VLAN20

VLAN No	VLAN Name
1	default
10	VLAN10
20	VLAN20

Figure 9.15 Adding new VLANs using the GUI

The next and last step is to configure the ports in such a way that they work in Access mode and belong to the appropriate VLANs.



FastEthernet0/1

Port Status: ☑ On

Bandwidth: ● 100 Mbps ○ 10 Mbps ☑ Auto

Duplex: ○ Half Duplex ● Full Duplex ☑ Auto

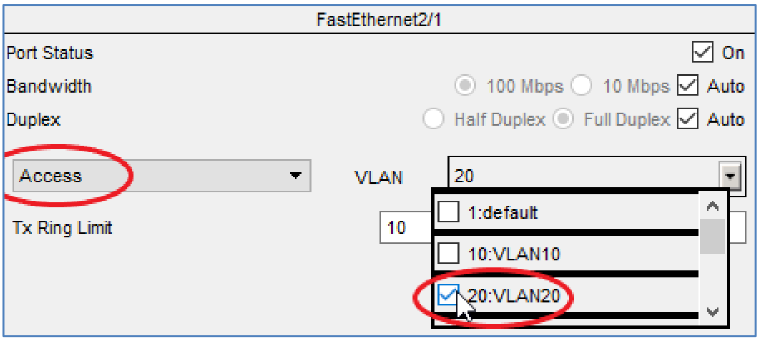
Access

VLAN: 10

Tx Ring Limit: 10

□ 1:default
☑ 10:VLAN10
□ 20:VLAN20

Figure 9.16 Assigning VLAN to the appropriate interface - VLAN10



FastEthernet2/1

Port Status: ☑ On

Bandwidth: ● 100 Mbps ○ 10 Mbps ☑ Auto

Duplex: ○ Half Duplex ● Full Duplex ☑ Auto

Access

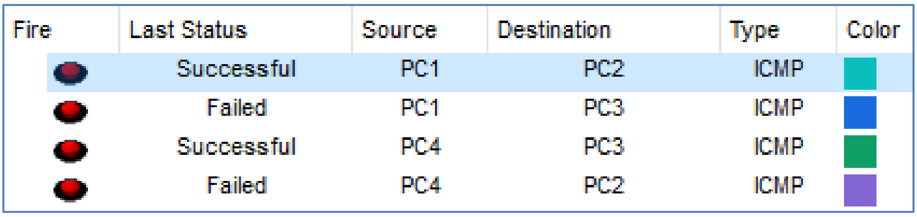
VLAN: 20

Tx Ring Limit: 10

□ 1:default
□ 10:VLAN10
☑ 20:VLAN20

Figure 9.17 Assigning VLAN to the appropriate interface - VLAN20

Similarly, we assign the other two interfaces to the respective ports. We can now test whether we are able to communicate in the area of one VLAN and whether we are able to connect to a computer located in another VLAN.



Fire	Last Status	Source	Destination	Type	Color
●	Successful	PC1	PC2	ICMP	🟦
●	Failed	PC1	PC3	ICMP	🟦
●	Successful	PC4	PC3	ICMP	🟩
●	Failed	PC4	PC2	ICMP	🟪

Figure 9.18 Test of communication in VLAN and between VLANs

As we might have expected - communication in the area of one VLAN is possible as if nothing has changed since the previous attempt, but communication between VLANs is blocked.

To illustrate how the port that is in **Trunk** mode works, we will add to our topology one more switch, connected to the first via optical fiber, and two additional computers located in **VLAN10** and **VLAN30**.

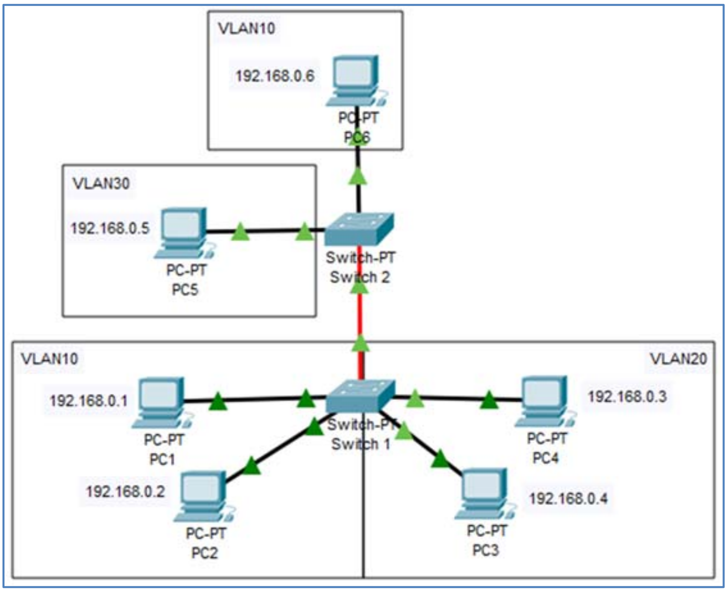
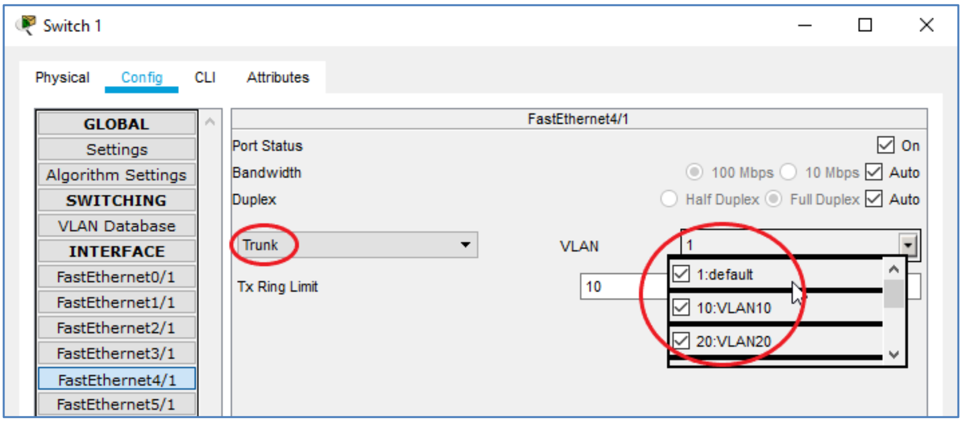


Figure 9.19 Topology with two switches to configure trunk connection (Example 9.2.2b.pkt)

After properly addressing two additional computers, we will configure the **Trunk** connection on the Switch **1**.



FastEthernet4/1

Port Status ☑ On

Bandwidth ● 100 Mbps ○ 10 Mbps ☑ Auto

Duplex ○ Half Duplex ● Full Duplex ☑ Auto

Trunk VLAN 1

☑ 1:default

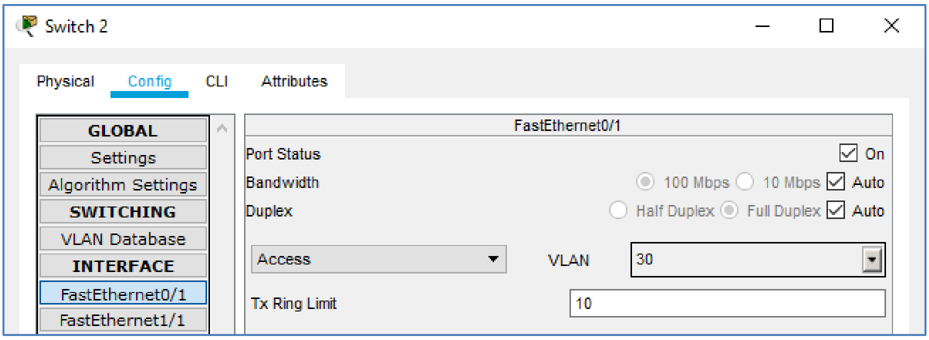
☑ 10:VLAN10

☑ 20:VLAN20

Tx Ring Limit 10

Figure 9.20 Trunk - Switch Connection Configuration 1

When you select **trunk** on the appropriate interface, all existing VLANs on the switch are automatically assigned to this port. We proceed to the configuration of **Switch 2**. We first create a database of VLANs - **VLAN10** and **VLAN30**, and then assign them to ports operating in **Access** mode.



FastEthernet0/1

Port Status ☑ On

Bandwidth ● 100 Mbps ○ 10 Mbps ☑ Auto

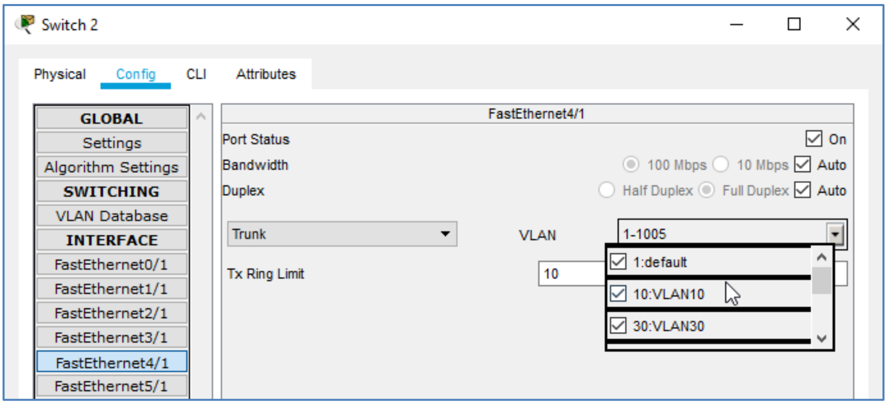
Duplex ○ Half Duplex ● Full Duplex ☑ Auto

Access VLAN 30

Tx Ring Limit 10

Figure 9.21 Port assignment to the corresponding VLAN - VLAN30

The second port leading to PC6, but assigned to VLAN10, should look similar. Now let's deal with the connection to The **Switch 1**. Let's set this port as a **Trunk** port and try to achieve communication between PC6 and PC1.



Switch 2

Physical | Config | CLI | Attributes

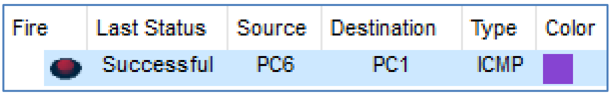
GLOBAL
Settings
Algorithm Settings
SWITCHING
VLAN Database
INTERFACE
FastEthernet0/1
FastEthernet1/1
FastEthernet2/1
FastEthernet3/1
FastEthernet4/1
FastEthernet5/1

FastEthernet4/1

Port Status ☑ On
Bandwidth ● 100 Mbps ○ 10 Mbps ☑ Auto
Duplex ○ Half Duplex ● Full Duplex ☑ Auto
Trunk VLAN 1-1005
Tx Ring Limit 10

☑ 1:default
☑ 10:VLAN10
☑ 30:VLAN30

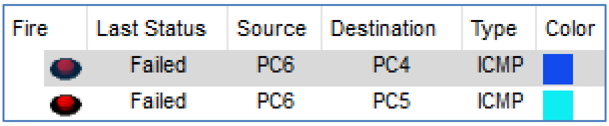
Figure 9.22 Trunk Connection Configuration - Switch 2



Fire	Last Status	Source	Destination	Type	Color
●	Successful	PC6	PC1	ICMP	🟪

Figure 9.23 Communication between PC6 and PC1

Everything works as it should, but when we try to communicate between VLANs again, we will not achieve a positive result, because we have not configured any router that would allow it.



Fire	Last Status	Source	Destination	Type	Color
●	Failed	PC6	PC4	ICMP	🟦
●	Failed	PC6	PC5	ICMP	🟦

Figure 9.24 Communication between VLANs

9.2.3 Enabling Communication Between VLANs

To enable communication between different virtual networks that have been created on switches, you need to add a router to our topology and connect it to one of the switches, for example the **Switch 2**, and configure the router with a function that we have already discussed – the so-called sub-interfaces. If we did not do this, we would have to create one connection between the switch and the router for each existing VLAN, which is neither practical nor simpler.

The topology that we will use to show how routing between VLANs works will look similar to the previous ones, but we will attach the router to the **Switch 2** and change the network addressing. Now each VLAN will have its own part of the **192.168.0.0/16** network due to the fact that you have to specify different default gateways on computers.

The first available network address will be the default gateway address, e.g. for the **192.168.10.0/24** network it will be **192.168.10.1.**

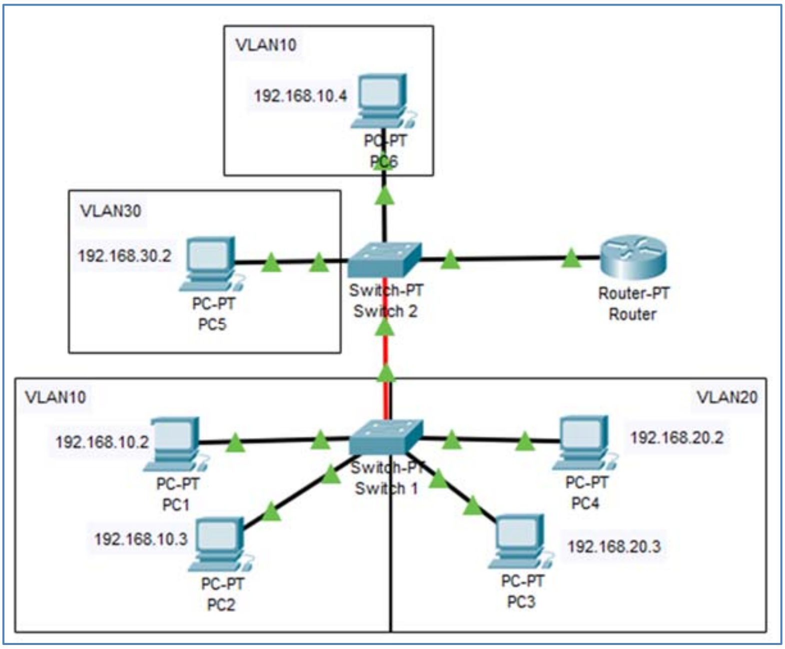
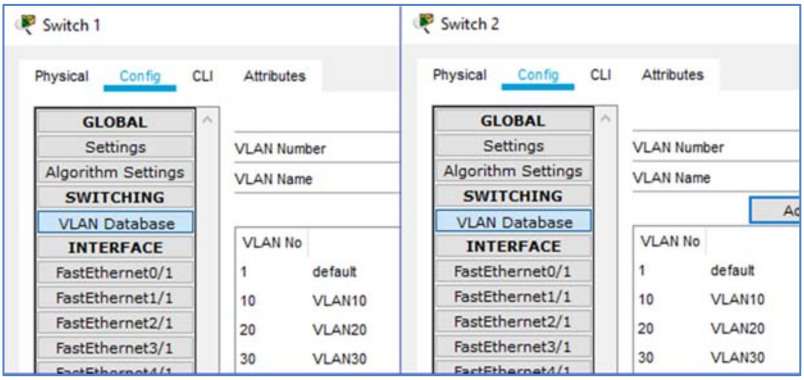


Figure 9.25 Topology for configuring Router on a stick (Example 9.2.3.pkt)

The first step to configure routing between virtual networks is to configure the switches themselves – you need to agree on **vlan databases** so that both switches have the same VLANs. This will be needed later due to the fact that on **trunk** links on both sides must be configured permission to move the same VLANs.



Switch 1

VLAN Number

VLAN Name

VLAN No	
1	default
10	VLAN10
20	VLAN20
30	VLAN30

Switch 2

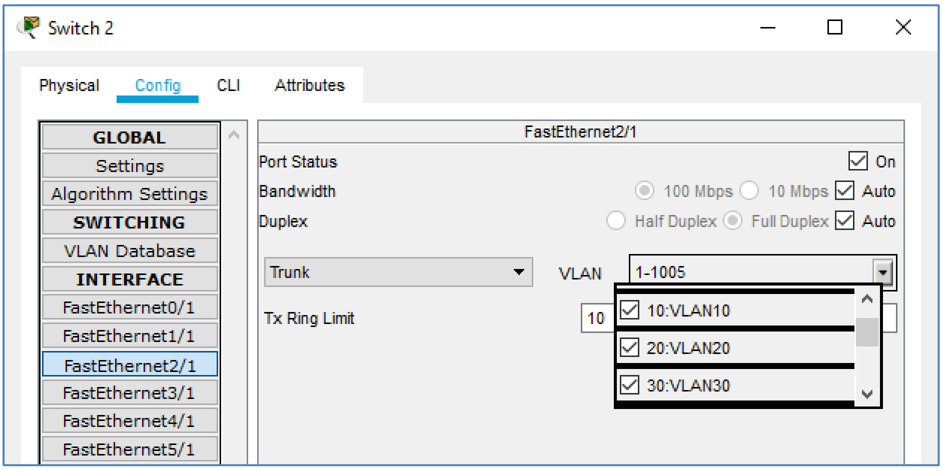
VLAN Number

VLAN Name

VLAN No	
1	default
10	VLAN10
20	VLAN20
30	VLAN30

Figure 9.26 VLAN Database - Switch 1 and Switch 2

When the VLAN base agrees on both switches, **Trunk** connections should also be allowed to move the same VLANs from two sides (on **Switch 1** and on **Switch 2**). It's time to switch to Switch 2 and set up one more **Trunk** connection – between the switch and the router.



FastEthernet2/1

Port Status ☑ On

Bandwidth ● 100 Mbps ○ 10 Mbps ☑ Auto

Duplex ○ Half Duplex ● Full Duplex ☑ Auto

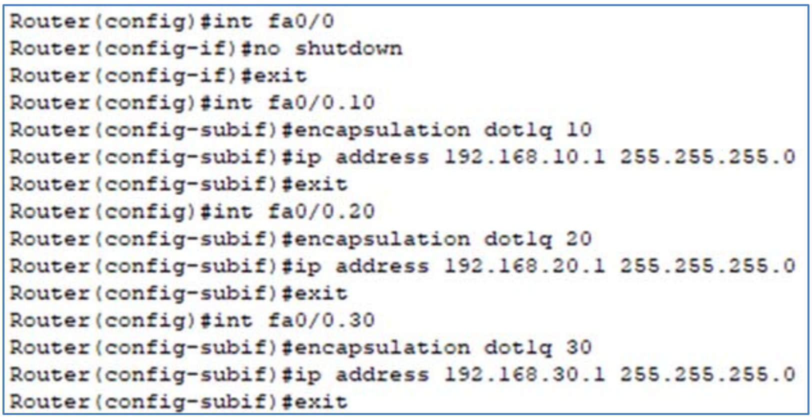
Trunk VLAN 1-1005

Tx Ring Limit 10

☑ 10:VLAN10
☑ 20:VLAN20
☑ 30:VLAN30

Figure 9.27 Setting up another Trunk - Switch 2 connection

We can now configure the Interface of the Fa0/0 router using commands to create three sub-interfaces – one for each existing VLAN. The next step is to execute the appropriate command on each **`encapsulation dot1q <VLAN number >`** (thanks to which the router will know how to mark, i.e. tag each frame from a given sub-interface) and assign the appropriate IP addresses for each sub-interface as shown in the figure below.



```
Router(config)#int fa0/0
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#int fa0/0.10
Router(config-subif)#encapsulation dot1q 10
Router(config-subif)#ip address 192.168.10.1 255.255.255.0
Router(config-subif)#exit
Router(config)#int fa0/0.20
Router(config-subif)#encapsulation dot1q 20
Router(config-subif)#ip address 192.168.20.1 255.255.255.0
Router(config-subif)#exit
Router(config)#int fa0/0.30
Router(config-subif)#encapsulation dot1q 30
Router(config-subif)#ip address 192.168.30.1 255.255.255.0
Router(config-subif)#exit
```

Figure 9.28 Configuring Router-on-a-stick

It remained for us to carry out tests of communication between VLANs. Through the configuration process, we could already understand how the idea of **Router on a stick** works, in which the router is the default gateway for each VLAN and it is he who performs routing between VLANs. Until this router is included in the network, VLANs cannot communicate with each other.

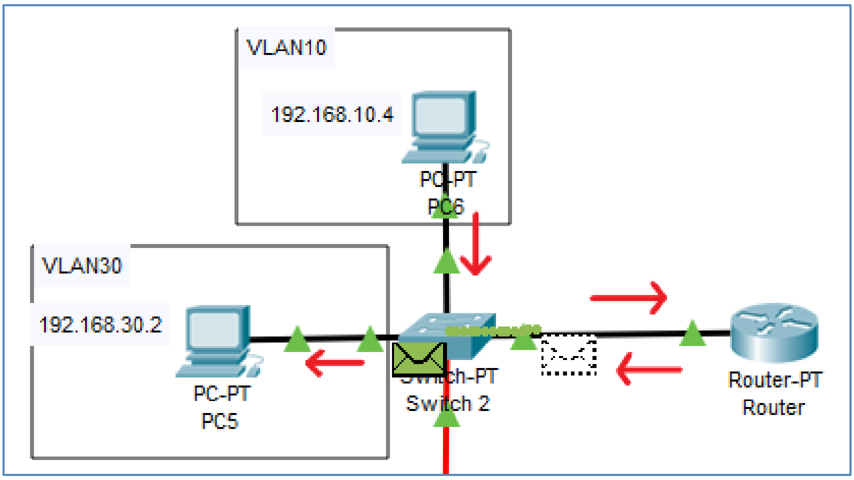
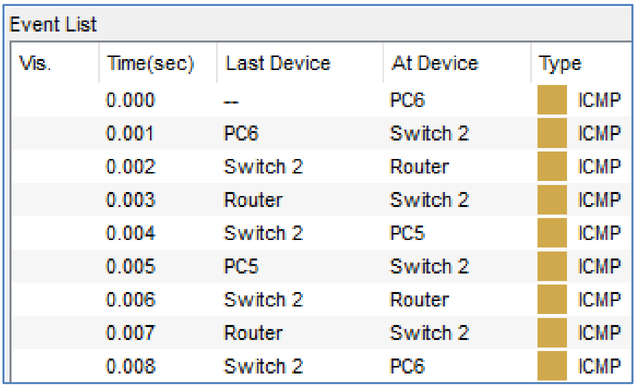


Figure 9.29 Traffic from VLAN10 to VLAN30

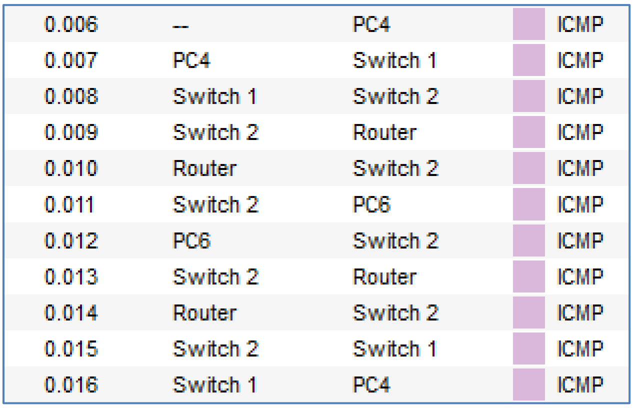
The figure above shows how the communication between VLANs looks like. Using the PING commands, you can confirm the path traveled by the packet, which is shown in the figure below.



Event List

Vis.	Time(sec)	Last Device	At Device	Type
	0.000	--	PC6	ICMP
	0.001	PC6	Switch 2	ICMP
	0.002	Switch 2	Router	ICMP
	0.003	Router	Switch 2	ICMP
	0.004	Switch 2	PC5	ICMP
	0.005	PC5	Switch 2	ICMP
	0.006	Switch 2	Router	ICMP
	0.007	Router	Switch 2	ICMP
	0.008	Switch 2	PC6	ICMP

Figure 9.30 PING between VLAN10 and VLAN30



0.006	--	PC4	🟪	ICMP
0.007	PC4	Switch 1	🟪	ICMP
0.008	Switch 1	Switch 2	🟪	ICMP
0.009	Switch 2	Router	🟪	ICMP
0.010	Router	Switch 2	🟪	ICMP
0.011	Switch 2	PC6	🟪	ICMP
0.012	PC6	Switch 2	🟪	ICMP
0.013	Switch 2	Router	🟪	ICMP
0.014	Router	Switch 2	🟪	ICMP
0.015	Switch 2	Switch 1	🟪	ICMP
0.016	Switch 1	PC4	🟪	ICMP

Figure 9.31 PING between VLAN20 and VLAN10

9.3 Configuring Cisco Switches in the IOS

9.3.1 Basic Information

In order to start configuring the switch in a real environment, you will need to have any computer with a COM port (**RS232**), a **console** cable and, of course, the switch itself. We connect the computer to the switch in the same way as in the case of a router, so we will not describe it here again.

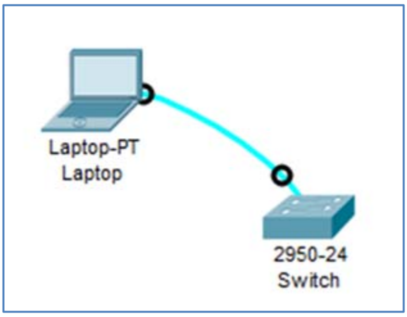
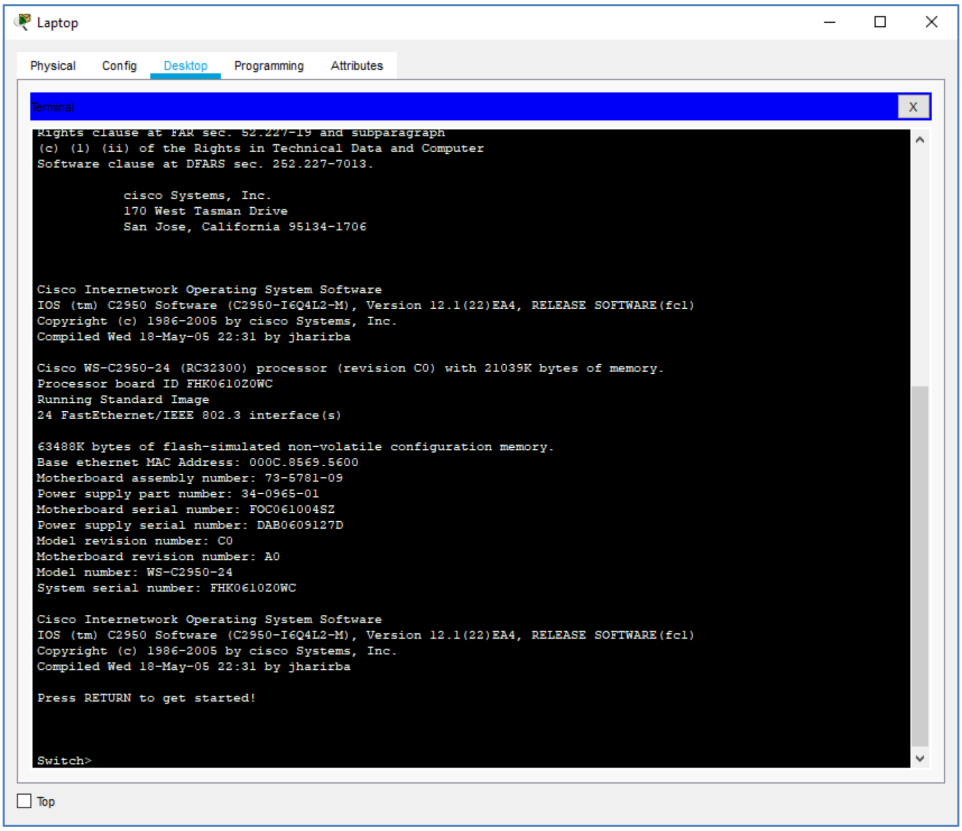


Figure 9.32 Correct console connection

The connection to the switch is established in the same way as in the case of a router, using the **Terminal** application. In the configuration window that appears after starting **Terminal**, do not change any parameters and click **OK**.



```
Rights clause at FAR sec. 52.227-19 and subparagraph
(c) (1) (ii) of the Rights in Technical Data and Computer
Software clause at DFARS sec. 252.227-7013.

           cisco Systems, Inc.
           170 West Tasman Drive
           San Jose, California 95134-1706



Cisco Internetwork Operating System Software
IOS (tm) C2950 Software (C2950-I6Q4L2-M), Version 12.1(22)EA4, RELEASE SOFTWARE(fc1)
Copyright (c) 1986-2005 by cisco Systems, Inc.
Compiled Wed 18-May-05 22:31 by jharirba

Cisco WS-C2950-24 (RC32300) processor (revision C0) with 21039K bytes of memory.
Processor board ID FHK0610Z0WC
Running Standard Image
24 FastEthernet/IEEE 802.3 interface(s)

63488K bytes of flash-simulated non-volatile configuration memory.
Base ethernet MAC Address: 000C.8569.5600
Motherboard assembly number: 73-5781-09
Power supply part number: 34-0965-01
Motherboard serial number: FOC061004SZ
Power supply serial number: DAB0609127D
Model revision number: C0
Motherboard revision number: A0
Model number: WS-C2950-24
System serial number: FHK0610Z0WC

Cisco Internetwork Operating System Software
IOS (tm) C2950 Software (C2950-I6Q4L2-M), Version 12.1(22)EA4, RELEASE SOFTWARE(fc1)
Copyright (c) 1986-2005 by cisco Systems, Inc.
Compiled Wed 18-May-05 22:31 by jharirba

Press RETURN to get started!


Switch>
```

Figure 9.33 Correctly log into the switch

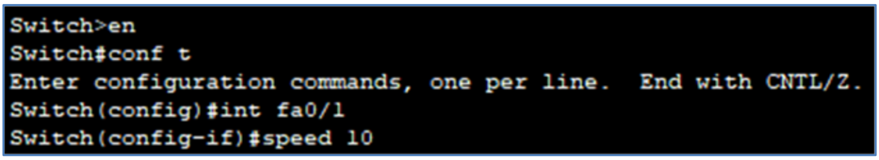
9.3.2 Basic Switch Configuration Modes

Unlike routers, switches do not have a wizard mode, so they must be configured only manually. Basic functions such as navigating the console, using help, and even checking the status of the device look analogous to routers, so they will not be described again.

9.3.3 Interface Configuration

The configuration of a given interface is analogous to the configuration of the interface in a router. First, we go into the global configuration mode, and then using the interface command **interface <type><number>** we configure the interface.

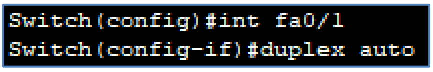
You can change the port bandwidth using the command **speed <bandwidth of Mb/s>**.



```
Switch>en
Switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#int fa0/1
Switch(config-if)#speed 10
```

Figure 9.34 Change interface bandwidth

Another interface parameter that we can configure is duplex mode. The mode is set with the command **duplex <mode>**, where the available modes are: **full, half** and **auto**.

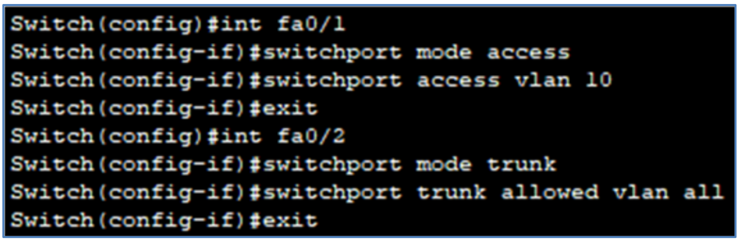


```
Switch(config)#int fa0/1
Switch(config-if)#duplex auto
```

Figure 9.35 Changing the duplex mode of the interface

Another option is to configure the interface to use VLANs. If the interface is to be in **Access** mode and only one VLAN is to belong to it, we use the **`switchport mode access`** command to set the interface mode to **`access`**, and then assign the interface to a specific VLAN with the switchport access VLAN command **`switchport access vlan [number]`**. On the other hand, if the interface is to work in **trunk** mode, instead of the two previous commands, enter **`switchport mode trunk`**. How to use these commands is described in the chapter on VLAN configuration.

To assign **VLANs** to the **Trunk** port we use the switchport trunk allowed vlan command **`switchport trunk allowed vlan [all or VLAN number]`**,, of course the default option is **all.**



```
Switch(config)#int fa0/1
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 10
Switch(config-if)#exit
Switch(config)#int fa0/2
Switch(config-if)#switchport mode trunk
Switch(config-if)#switchport trunk allowed vlan all
Switch(config-if)#exit
```

Figure 9.36 Changing the mode of operation of ports and assigning VLANs

9.3.4 VLAN Configuration

We will start by creating the following topology, on the example of which we will configure VLANs:

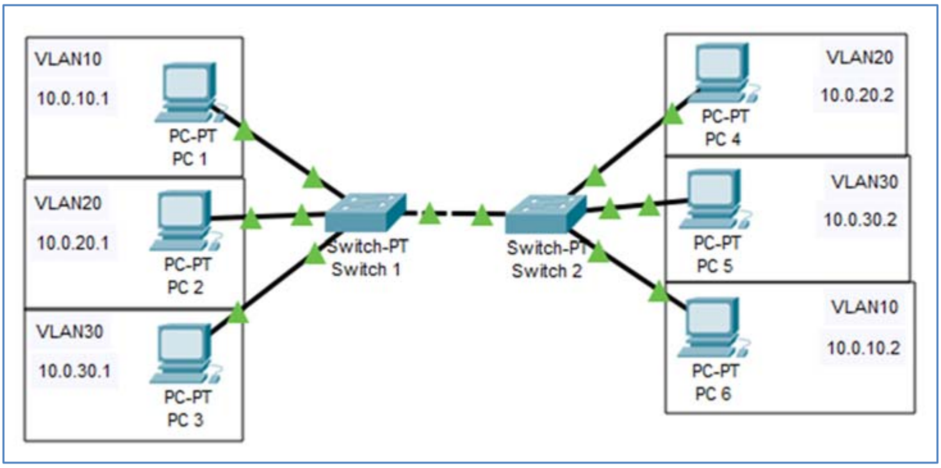
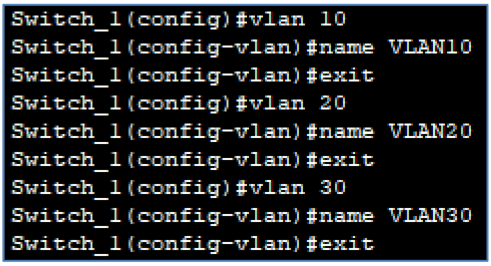


Figure 9.37 Topology for switch configuration (Example 9.3.4.pkt)

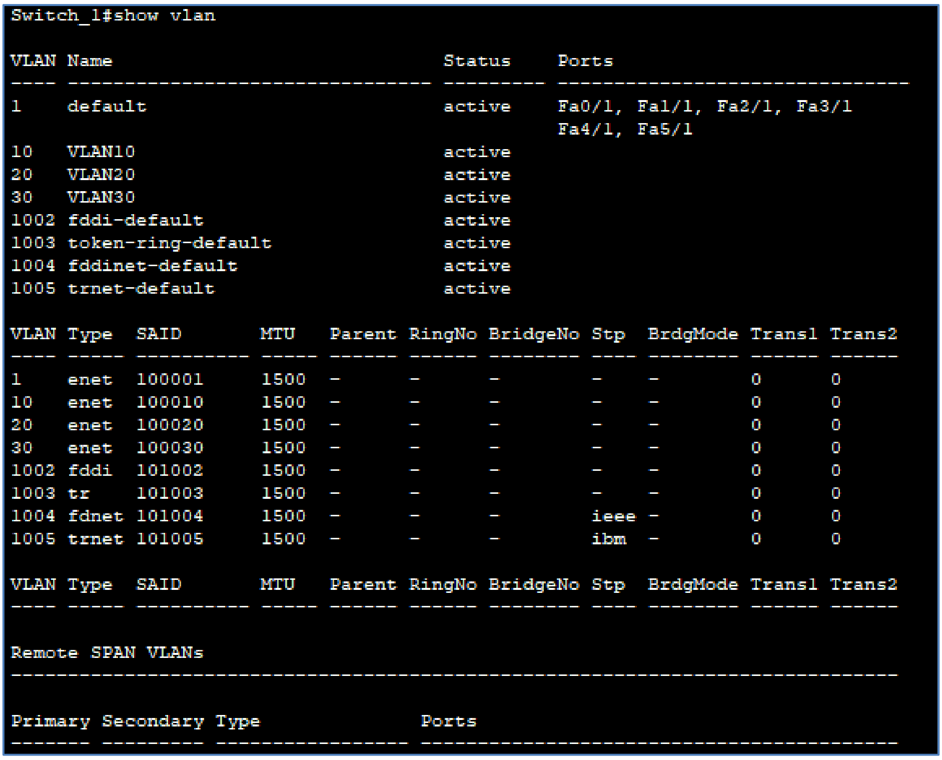
As you can see, the topology predicts the existence of three VLANs – VLAN 10, VLAN 20 and VLAN 30. So we'll start by creating them on the switches. To create a VLAN, we must first enter the **global configuration** mode and use the command with the syntax: **`vlan [number]`**.. We can then name the virtual network by using the **`name`** command. We return to the **global configuration** mode with the **`exit`** command, and then create the second necessary VLAN and the third. Repeat on the second switch as well.



```
Switch_1(config)#vlan 10
Switch_1(config-vlan)#name VLAN10
Switch_1(config-vlan)#exit
Switch_1(config)#vlan 20
Switch_1(config-vlan)#name VLAN20
Switch_1(config-vlan)#exit
Switch_1(config)#vlan 30
Switch_1(config-vlan)#name VLAN30
Switch_1(config-vlan)#exit
```

Figure 9.38 Create VLANs on the switch

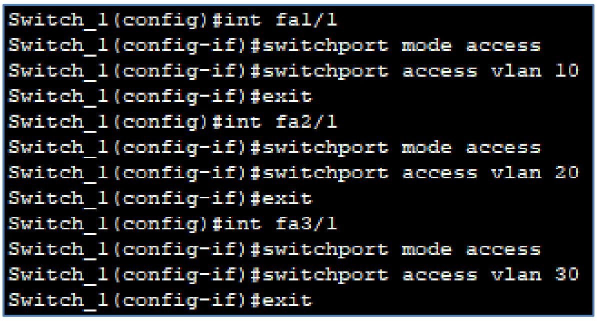
If we do not give VLAN a name, it will get the default name, in the above case it would be VLAN0010. VLANs can also be deleted, we do this by typing the command **`no vlan [number]`** in the **global configuration** mode. Now let's check if the corresponding VLANs have been created correctly. To do this, we need to enter **privileged mode** and then use the **`show vlan`** command.



```
Switch_1#show vlan

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/1, Fa1/1, Fa2/1, Fa3/1
                                                Fa4/1, Fa5/1
10   VLAN10                           active
20   VLAN20                           active
30   VLAN30                           active
1002 fddi-default                     active
1003 token-ring-default               active
1004 fddinet-default                  active
1005 trnet-default                    active

VLAN Type  SAID       MTU   Parent RingNo BridgeNo Stp  BrdgMode Trans1 Trans2
---- ----- ---------- ----- ------ ------ -------- ---- -------- ------ ------
1    enet  100001     1500  -      -      -        -    -        0      0
10   enet  100010     1500  -      -      -        -    -        0      0
20   enet  100020     1500  -      -      -        -    -        0      0
30   enet  100030     1500  -      -      -        -    -        0      0
1002 fddi  101002     1500  -      -      -        -    -        0      0
1003 tr    101003     1500  -      -      -        -    -        0      0
1004 fdnet 101004     1500  -      -      -        ieee -        0      0
1005 trnet 101005     1500  -      -      -        ibm  -        0      0

VLAN Type  SAID       MTU   Parent RingNo BridgeNo Stp  BrdgMode Trans1 Trans2
---- ----- ---------- ----- ------ ------ -------- ---- -------- ------ ------

Remote SPAN VLANs
------------------------------------------------------------------------------

Primary Secondary Type              Ports
------- --------- ----------------- ------------------------------------------
```

Figure 9.39 Show vlan command on a switch

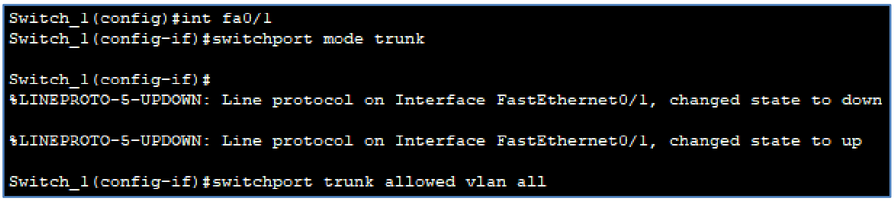
We can find here information about VLANs that were created on the switch and what ports are assigned to the VLAN data. We will start by assigning those interfaces that connect computers to the switch. So we enter the **global configuration** mode, and then the interface, in the way already described above. These interfaces belong to individual VLANs and connect to end devices, so they will work in **Access** mode.



```
Switch_1(config)#int fa1/1
Switch_1(config-if)#switchport mode access
Switch_1(config-if)#switchport access vlan 10
Switch_1(config-if)#exit
Switch_1(config)#int fa2/1
Switch_1(config-if)#switchport mode access
Switch_1(config-if)#switchport access vlan 20
Switch_1(config-if)#exit
Switch_1(config)#int fa3/1
Switch_1(config-if)#switchport mode access
Switch_1(config-if)#switchport access vlan 30
Switch_1(config-if)#exit
```

Figure 9.40 Changing the operating modes on the switch interfaces

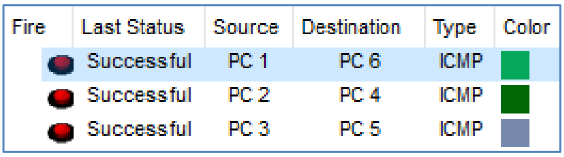
Similarly, it should look on the second switch with the corresponding change of interfaces. Next, proceed to change the operating mode of the interface, connecting both switches. It connects two network devices and will transmit data from several VLANs, so it will work in **Trunk** mode.



```
Switch_1(config)#int fa0/1
Switch_1(config-if)#switchport mode trunk

Switch_1(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to down

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up

Switch_1(config-if)#switchport trunk allowed vlan all
```

Figure 9.41 Change the interface mode to Trunk

To perform connection tests between once whether computers in the same VLANs can communicate with each other.



Fire	Last Status	Source	Destination	Type	Color
●	Successful	PC 1	PC 6	ICMP	🟩
●	Successful	PC 2	PC 4	ICMP	🟩
●	Successful	PC 3	PC 5	ICMP	🟦

Figure 9.42 PING between VLANs

9.3.5 Configuration of Virtual Terminals (Telnet, SSH)

In the case of switches, as for routers, they can be configured remotely from LAN or even VAN via virtual terminals such as **Telnet** or **SSH**. Now let's create the following topology.

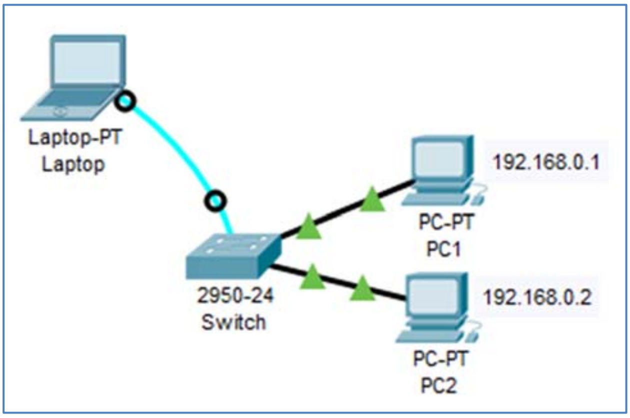
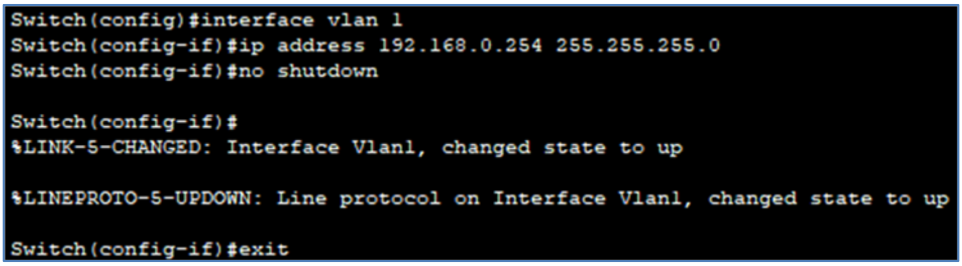


Figure 9.43 Topology to configure remote configuration

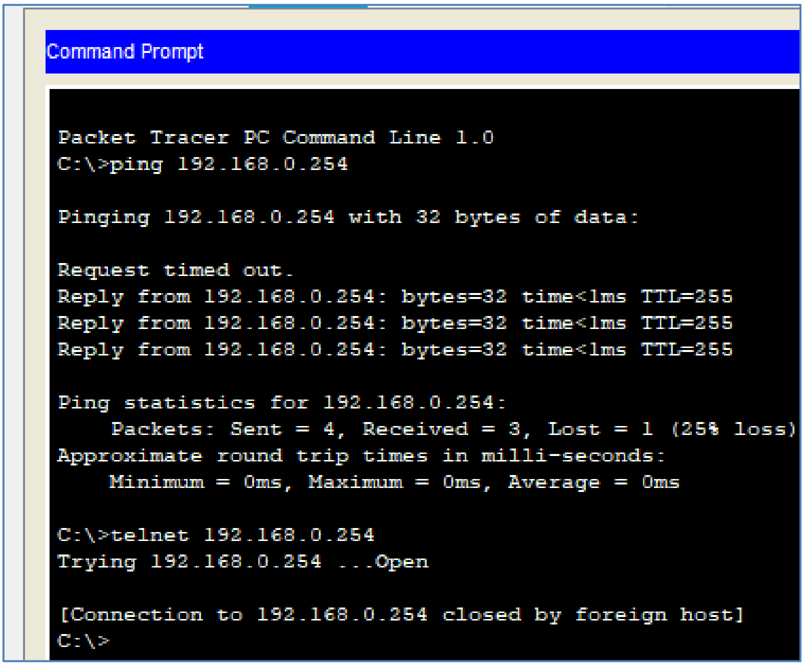
To configure a switch to connect to it via Telnet or SSH, you must first give it an IP address so that it is visible on the network. Configure the **VLAN1** interface and give it an IP address, e.g. **192.168.0.254/24**.



```
Switch(config)#interface vlan 1
Switch(config-if)#ip address 192.168.0.254 255.255.255.0
Switch(config-if)#no shutdown

Switch(config-if)#
%LINK-5-CHANGED: Interface Vlan1, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed state to up

Switch(config-if)#exit
```

Figure 9.44 Giving an IP address for VLAN 1

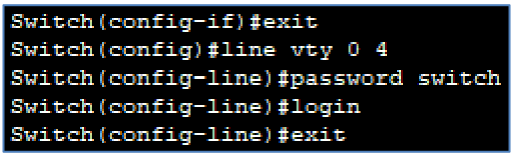
After such a configuration, computers in the topology will be able to communicate with the switch, e.g. using the PING command, however, if we try to connect to it using Telnet, we will receive the message **`Connection to 192.168.0.254 closed by foreign host`**, as you can see in the figure below.



```
Packet Tracer PC Command Line 1.0
C:\>ping 192.168.0.254

Pinging 192.168.0.254 with 32 bytes of data:

Request timed out.
Reply from 192.168.0.254: bytes=32 time<1ms TTL=255
Reply from 192.168.0.254: bytes=32 time<1ms TTL=255
Reply from 192.168.0.254: bytes=32 time<1ms TTL=255

Ping statistics for 192.168.0.254:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss)
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>telnet 192.168.0.254
Trying 192.168.0.254 ...Open

[Connection to 192.168.0.254 closed by foreign host]
C:\>
```

Figure 9.45 Attempt to communicate with a switch

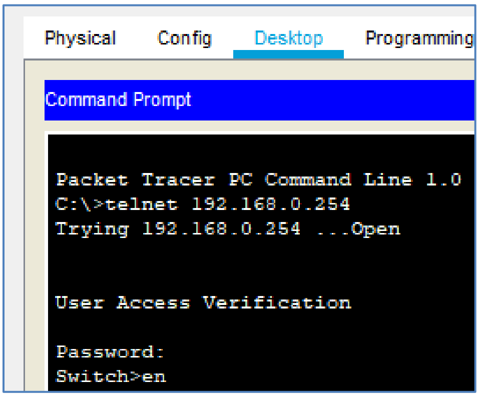
To connect to the switch, we must enable the ability to remotely connect via Telnet, as well as enable authorization by setting a password. We do it in the same way as on a router.



```
Switch(config-if)#exit
Switch(config)#line vty 0 4
Switch(config-line)#password switch
Switch(config-line)#login
Switch(config-line)#exit
```

Figure 9.46 Activating switch access using virtual terminals

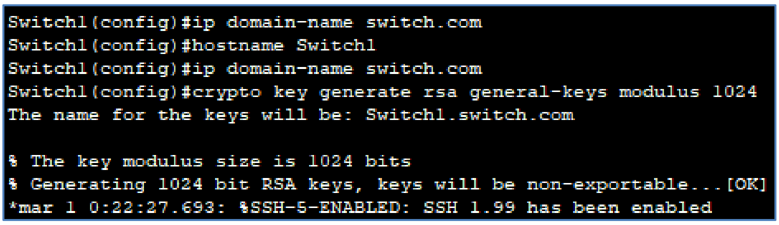
Now we are able to connect to the switch using the **Telnet** protocol:



```
Packet Tracer PC Command Line 1.0
C:\>telnet 192.168.0.254
Trying 192.168.0.254 ...Open


User Access Verification

Password:
Switch>en
```

Figure 9.47 Positive attempt to access the switch using the Telnet protocol

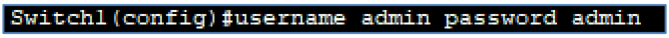
Once we have access via Telnet, we can configure a more secure connection – a connection using the **SSH** protocol. To do this, you need to rename the **switch** (due to the fact that Switch is the default name of the switch and SSH is a secured protocol, so it does not allow default values for names, among other things). After that, configure the IP domain and generate RSA keys, which are already described in the section on configuring SSH access to the router.



```
Switch1(config)#ip domain-name switch.com
Switch1(config)#hostname Switch1
Switch1(config)#ip domain-name switch.com
Switch1(config)#crypto key generate rsa general-keys modulus 1024
The name for the keys will be: Switch1.switch.com

% The key modulus size is 1024 bits
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]
*mar 1 0:22:27.693: %SSH-5-ENABLED: SSH 1.99 has been enabled
```

Figure 9.48 Configure SSH switch access

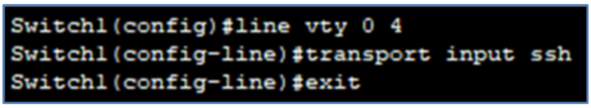
You must also create a user (login and password) to which you will be able to log in.



```
Switch1(config)#username admin password admin
```

Figure 9.49 Create a user on a switch

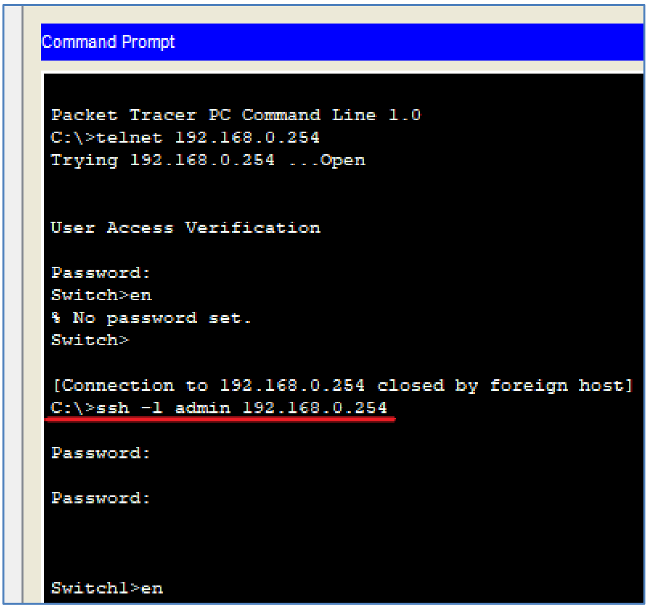
Next, you need to perform permission on the switch for SSH connections.



```
Switch1(config)#line vty 0 4
Switch1(config-line)#transport input ssh
Switch1(config-line)#exit
```

Figure 9.50 Permission for internal SSH connections

Now you need to perform a login test (access using SSH) from the computer to the switch.



```
Packet Tracer PC Command Line 1.0
C:\>telnet 192.168.0.254
Trying 192.168.0.254 ...Open


User Access Verification

Password:
Switch>en
% No password set.
Switch>

[Connection to 192.168.0.254 closed by foreign host]
C:\>ssh -l admin 192.168.0.254

Password:

Password:



Switch1>en
```

Figure 9.51 SSH access to the switch

9.3.6 REP Protocol

Cisco RESILIENT Ethernet Protocol (**Cisco REP**) is a Cisco protocol used in networks known as **Carrier Ethernet**. Carrier Ethernet is a technology used to build large WAN structures based on the Ethernet standard. This technology allows **ISR** providers and operators to build large-area, multi-branch corporate networks to provide standardized services. The primary goals of Carrier Ethernet are standardization, scalability, management, reliability, and quality of service.

The main features of the REP protocol are:

- the protocol is owned by Cisco,
- it is mainly used in Metro Ethernet networks based on ring topology,
- it allows you to prevent the formation of a second layer loop,
- it is faster than the STP protocol, it can provide short network convergence times (about 50 mS – 150 mS).

9.3.6.1 *Purpose of REP*

The main purpose of REP is to prevent Layer 2 loops in physical and logical topologies of the following types:

- point-to-point,
- star,
- bus,
- ring.

9.3.6.2 *Basic Concepts of REP*

REP segment – a chain of ports connected to each other and marked (configured) with one and the same identifier, the so-called **segment ID**. The ID segment is an integer from 1 d o1024.

A **REP edge switch** is a switch located at the edge of a REP segment. Edge port of the REP segment (edge port) – a **port of the edge switch**, located inside the REP segment.

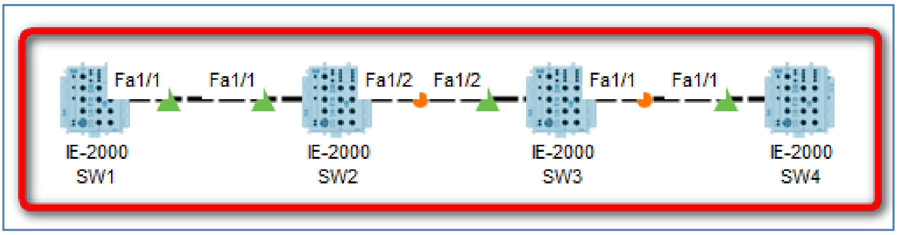


Figure 9.52 An example of a REP segment in a point-to-point topology.

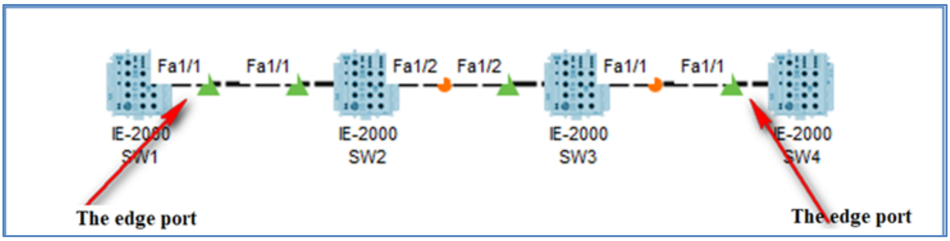


Figure 9.53 An example of a REP segment in a point-to-point topology.

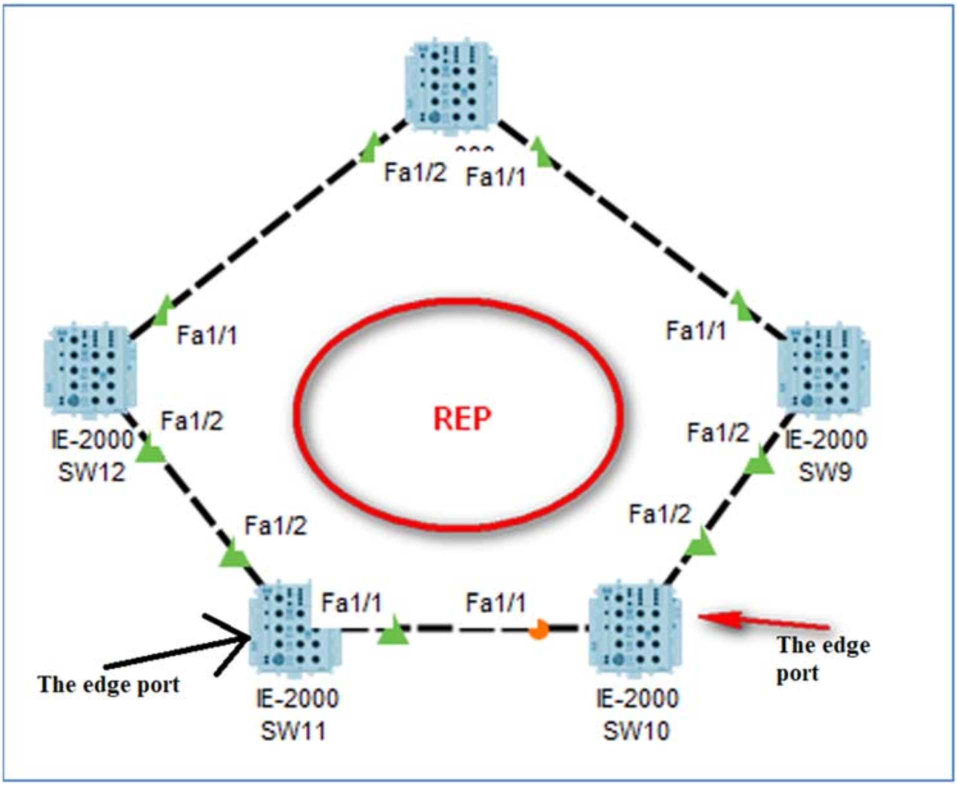


Figure 9.54 Sample REP segment in Ring topology

Setting up a REP segment is very simple. On each of the ports inside the **REP** segment, use the following commands:

```
SW1 (config-if)# sw mod trunk
SW1 (config-if)# rep segment N
```

where **N** is an integer **between 1 and 1024**, denoting the segment ID. On each switch, execute the following commands.

```
en
conf t
interface fa1/1
sw mod trunk
rep segment 1
interface fa1/2
sw mod trunk
rep segment 1
```