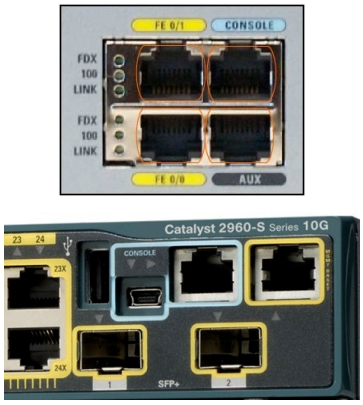
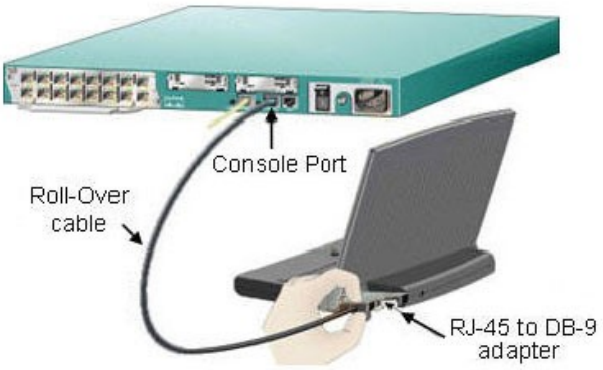
Redes de Computadores (RCOMP) – 2017/2018 Laboratory Class Script – PL06	
• IPv4 addressing and static routeing. • Command Line Interface (CLI) management of network devices. • VLAN support on Cisco routers – switching module. • VLAN support on Cisco routers – sub interfaces. • Spanning Tree Protocol (STP).	• Cisco Packet Tracer

1. Command Line Interface (CLI) management of network devices

The basic tool to manage a network device is the command line interface (CLI), other available tools like a WEB interface or SNMP (Simple Network Management Protocol) only provide a small subset of features available at the command line. Also, management tools operating through the network are not available when the device is not yet configured.

Access to the command line interface can be achieved either through the network (when device network settings are already in place) or through a **console connection** for a fresh new device:





In most network devices, a console connection can be accomplished by a serial line cable, traditionally to an RS-232 port, nowadays a USB port. A dedicated text terminal device can be used for command line interaction, but also a terminal emulator (e.g. PuTTY or Tera Term) running on a standard workstation or laptop.

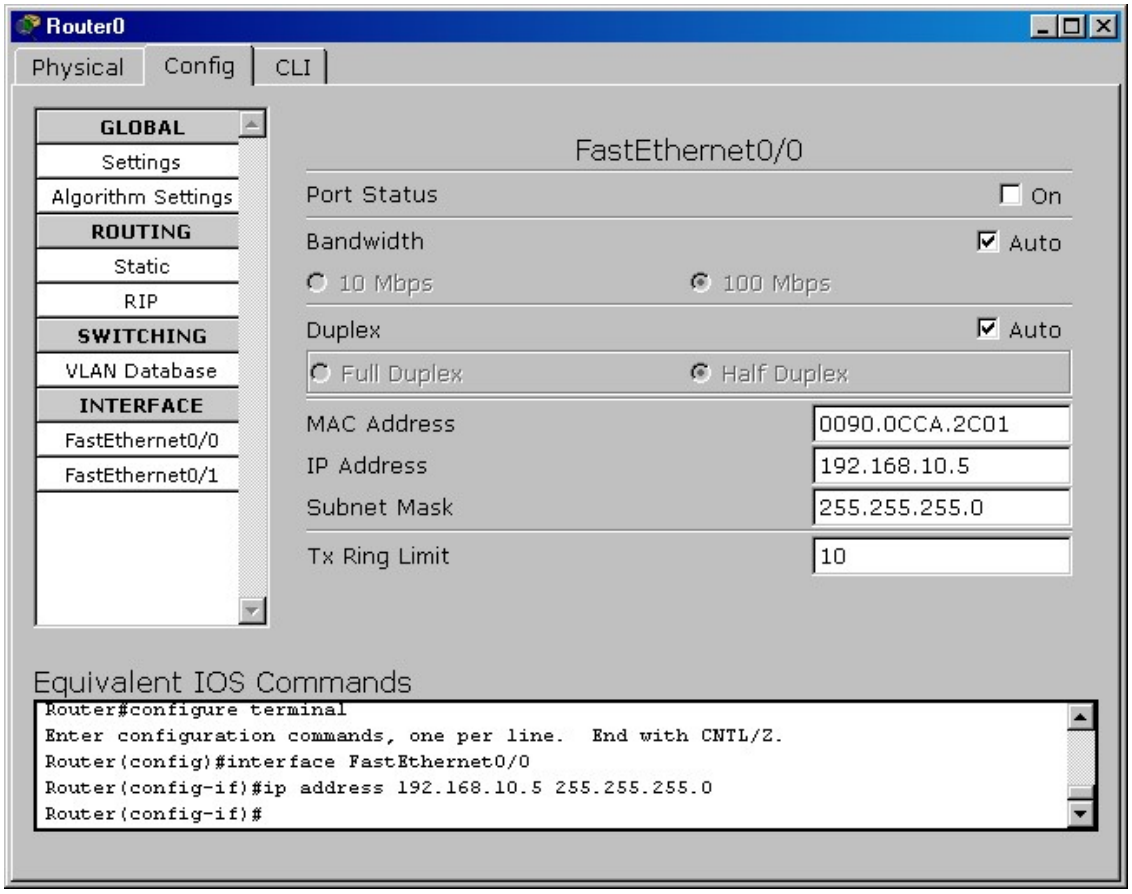
After initial settings at the console, including network basic configuration and remote access services, the command line interface will then be reachable through the network itself by using the TELNET protocol (unsecure) or the SSH (Secure Shell) protocol. Most terminal emulators support both these protocols in addition to the direct serial line connection.

Cisco Packet Tracer allows beginners to configure devices through friendly forms and thus avoid the CLI. However, the ultimate goal is to gradually start using the command line interface (on real devices there are no forms available). On Cisco Packet Tracer, for each management action performed using the form, the command line equivalent is presented on a window at the bottom of the form (Equivalent IOS Commands).

Whenever the student feels ready, he can start switching to CLI by selecting the CLI tab on the device´s form.

Be aware that management actions available at Packet Tracer forms are only a small subset of all actions Packet Tracer indeed supports for that device in CLI. Also be aware, features supported by Packet Tracer devices don´t include all features available in the real device.

The following sample image shows the Packet Tracer form for configuring a router's interface, here we can see that this action corresponds to two CLI commands, one for selecting the interface to be configured and another to set the IPv4 address and network mask for that interface.



1.1. CLI command modes

The CLI is privilege-level or command-mode oriented, there are several command modes, for each a different set of commands are available, the terminal prompt indicates the mode we are in.

The lowest privilege-level is zero (user EXEC mode) and the prompt will be the greater-than symbol, **>**. The most significant command at this level is **enable**, the **enable** command enters a higher privilege-level, the default is privilege-level 15 (privileged EXEC mode), at this level the prompt is a hash symbol: **#**.

In **privileged EXEC mode**, all the device configuration and status can be queried, also, from this level we can enter **configuration mode** by using the **configure** command, the prompt will then be (config)#. If the configuration is to be performed from the terminal the full command should be **configure terminal**.

Although the **privileged EXEC mode** allows querying about any device feature status, it does not allow configuration changes, for that purpose we must enter **configuration mode**. There are also several configuration sub-modes, for instance at the previous image we can see the **interface** command was used in the **configuration mode** to enter the **interface configuration sub-mode**, represented by the (config-if)# prompt.

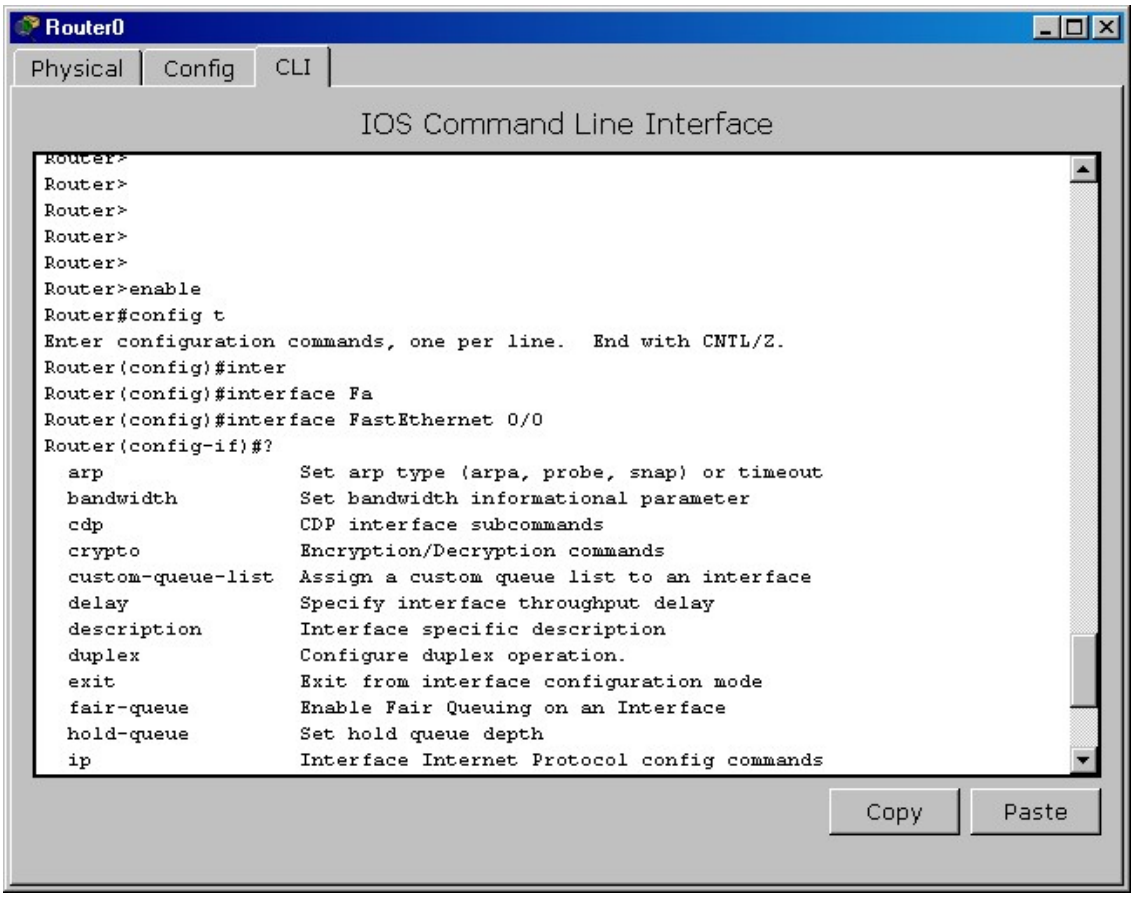
At any configuration mode, the **end** command can be used to leave back to the **privileged EXEC mode**. The **exit** command can be used in any mode to go back to the previous mode.

Most configuration commands can be reverted by repeating the exact same command line preceded by the word **no**. For instance, in interface configuration sub-mode **shutdown** to disable the interface and **no shutdown** to enable the interface.

1.2. CLI command typing

For command typing at CLI several useful features are available. The TAB key is used for complete, a partially written command or command argument can be completed by the system by pressing this key. As far as they are not ambiguous, commands and arguments can also be abbreviated, for instance, **en** instead of **enable**, or **conf t** instead of **configure terminal**.

Both before entering a command or on a command argument the question mark may be used to get assistance, the system will provide a list of available valid commands or arguments.



At the image above we start in **user EXEC mode** and use the **enable** command to enter **privileged EXEC mode**, then enter **configuration mode** by using the **config t** abbreviated command.

In configuration mode, TAB was used to complete both the **interface** command and also the command argument. At last, in interface configuration sub-mode, the question mark was used to display available commands.

2. VLAN support in Cisco routers

We already know end nodes can also be VLAN-aware and handle IEEE 802.1q frames, this allows a single network interface to operate as several independent interfaces, each one connected to a different VLAN. End nodes operating at layer three can then assign different networks IP address to each interface.

2.1. Switching module for Cisco routers

This is, in fact, a workaround for connecting VLANs to Cisco routers, it's a hardware module that contains a VLAN-aware layer two switch. Once attached to the router every VLAN defined on it will be known to the router.

The image above shows the Cisco HWIC-4ESW module that can be attached to some Cisco routers, for instance to the 2811 model in the image below:

This is a four ports layer two switch, not four independent network interfaces for the router, the router will not be able to use these ports directly, however, it will be able to use every VLAN we define on the switch as an independent network interface.

Packet Tracer forms can be used to configure the switch VLANs as with any other switch. Afterwards, new interfaces become available to the router, they can be referred to as **vlan ID**, where ID represents the used VLANID.

For instance, if we define a VLAN with VLANID=25 on the switch and assign it to some switch ports, we can then connect the router to that VLAN by using the following commands:

```
(config) interface vlan 25
(config-if) ip address 192.168.0.10 255.255.255.0
```

The router will be using IP address 192.168.0.10/24 on that VLAN.

2.2. Sub-interfaces

Although the switching hardware module can do the trick, it requires additional hardware. The same goal can be achieved with no additional hardware. Please note added hardware may have restrictions on its own, for instance, the described hardware module works only at 100 Mbps and is not available for all router models.

Cisco IOS (Cisco devices operating system) and similar systems allow the creation of sub-interfaces over existing physical interfaces. Sub-interfaces overlap the physical interface and multiple sub-interfaces over the same physical interface overlap each other. Each sub-interface can have its own IP address and is independent of others.

Sub-interfaces are identified by the physical interface name with a numeric suffix separated by a dot, first sub-interface will have suffix one and so on. For instance interfaces **FastEthernet 0/0.1**, **FastEthernet 0/0.2**, and **FastEthernet 0/0.3** are **FastEthernet 0/0** sub-interfaces.

We can use the **encapsulation dot1q** command to add VLAN IEEE 802.1q tagging to sub-interfaces, thus, providing traffic separation within the network physical connection. This command has one additional argument, the VLANID. Once applied the interface will receive only 802.1q frames with that VLANID, also, any frame sent through this interface will be an 802.1q frame carrying that VLANID.

Let's take an example: we want to use interface **FastEthernet 0/0** to connect the router to two VLANs with IDs 50 and 60, this can be achieved by the following commands:

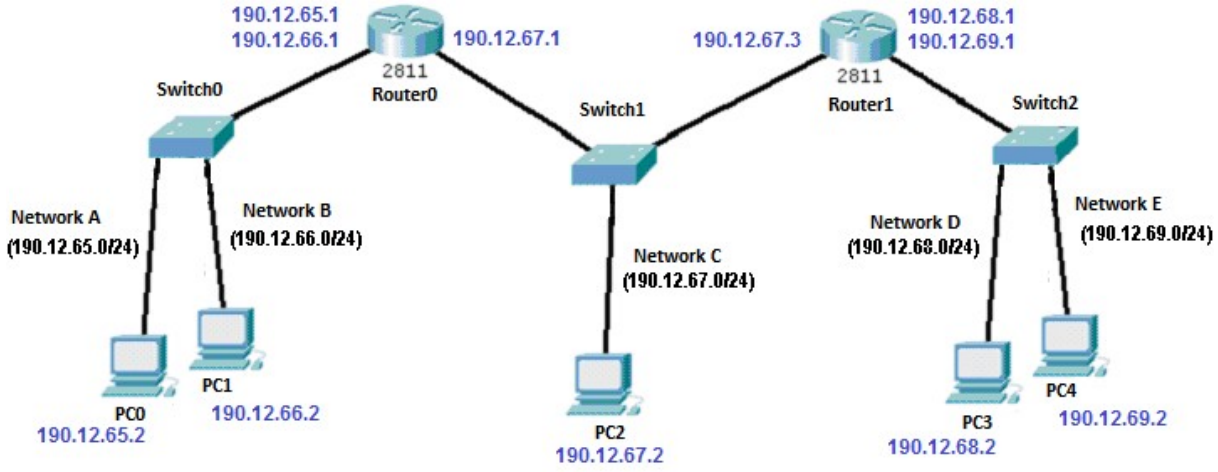
```
(config) interface FastEthernet 0/0.1
(config-subif) encapsulation dot1Q 50
(config-subif) ip address 195.1.0.20 255.255.255.0
(config-subif) no shutdown
(config-subif) exit

(config) interface FastEthernet 0/0.2
(config-subif) encapsulation dot1Q 60
(config-subif) ip address 190.5.0.2 255.255.255.0
(config-subif) no shutdown
(config-subif) exit
```

For this configuration to make sense, this router interface is supposed to be connected to the port of a switch with both the VLANs defined and assigned to that port (trunk-mode).

3. Use Packet Tracer to create the following scenario

We have here two routers and three physical networks (LANs), however, there are five IPv4 networks from A to E, networks A and B are different VLANs over one LAN and the same goes for networks D and E.



On Switch0 define VLANs with ids 5 and 6 for IPv4 networks A and B, plug the previously referred hardware switching module to Router0 and use it to connect Router0 to Switch0 in trunk-mode.

Switch1 is fully dedicated to IPv4 network C, and thus, no VLANs are required here.

On Switch2 define VLANs with ids 10 and 11 for IPv4 networks D and E, use sub-interfaces to connect Router1 to Switch2 in trunk-mode.

a) **Assign to nodes the shown IPv4 addresses.**

b) **Configure routeing (default-gateway at end nodes and routeing table at routers).**

c) **In simulation mode, test IPv4 connectivity by sending ICMP echo requests between all nodes.**

4. The Spanning Tree Protocol (STP)

One thing an ethernet network should never have is a loop, this is because ethernet and most layer two protocols lack any feature that would avoid frames from start circulating indefinitely around a loop. If that happens, every sent packet is added to the loop and never eliminated, thus, ultimately the network becomes unusable due to traffic overload.

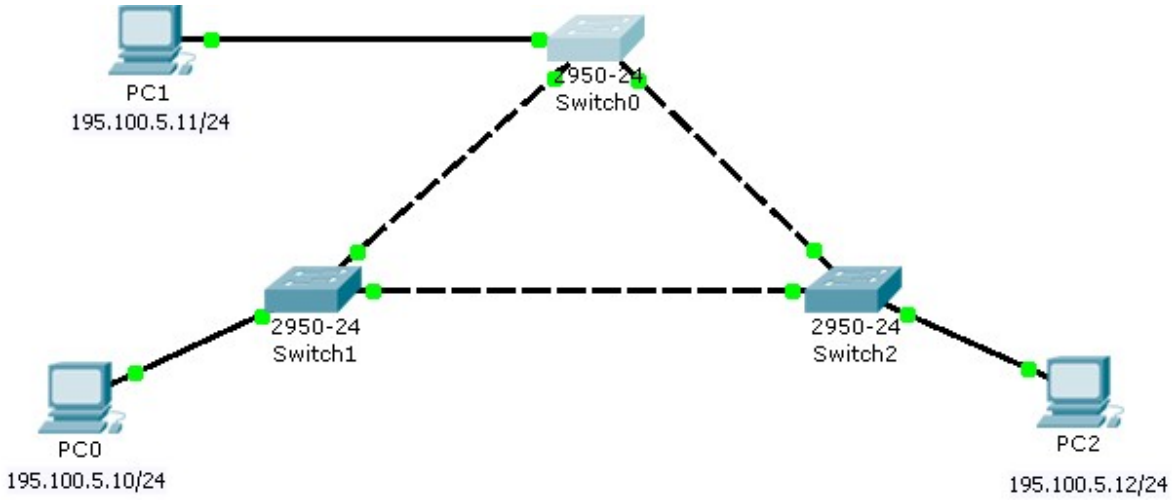
Nevertheless, alternative paths in a network are desirable, they can provide load balancing and fault tolerance.

For fault tolerance, STP can be used to disable existing loops. STP acts by detecting loops and temporarily disable some links to eliminate them, while STP is running if a currently active link fails a previously disable link is reactivated. This is, therefore, a failover mechanism, existing alternatives paths are not used at once, at each time there is only one active path.

STP does not provide load balancing, just failover. Load balancing over ethernet networks can be achieved by using Link Aggregation (port trunking).

Because loops have a major impact on ethernet networks, switches that support STP have it active by default.

5. Use Packet Tracer to create the following scenario



As we can see there is a loop, however, thanks to STP (enabled by default on all switches) the loop is broken by temporarily disabling one switch interface.

Now, let's see what would happen if STP was not doing its job.

a) Disable STP on all three switches.

Use CLI at each switch and, in configuration mode, run the **no spanning-tree vlan 1** command. This will disable STP on the default VLAN (VLANID=1).

```
(config) no spanning-tree vlan 1
```

b) Set the end nodes IPv4 addresses as shown in the image.

c) Enter simulation mode and send an ICMP echo request between two end nodes.

Check that the network becomes unusable.

d) Enable STP on all three switches

Enter real mode and enable STP. Use CLI at each switch and, in configuration mode, run the **spanning-tree vlan 1** command.

```
(config) spanning-tree vlan 1
```

Wait until STP breaks the loop (one switch connection will be on standby/orange).

e) Again in simulation mode, send an ICMP echo request between two end nodes.

Now the network is working ok, there are no active loops.

f) Testing fault tolerance (failover).

Enter real mode again. Disable one active connection (both ends green) between two switches. To simulate a connection failure disable one of the ports.

Wait until STP converges (redefines active ports).

You will see the port previously deactivated by STP is now reactivated.

g) Again in simulation mode, send an ICMP echo request between two end nodes.

Check that, again, the network is able to deliver packets between any end nodes.