**LAB # 01**

**AN OVERVIEW OF NETWORKING**

**OBJECTIVE**

Introduction to Networks with Cisco IOS CLI. Overview of real time and simulation mode.

**THEORY**

**COMPUTER NETWORKING**

Computer networking is the engineering discipline concerned with communication between computer systems or devices. Networking, routers, routing protocols, and networking over the public Internet have their specifications.

A computer network can be two computers connected:



A computer network can also consist of, and is usually made for, more than two computers:



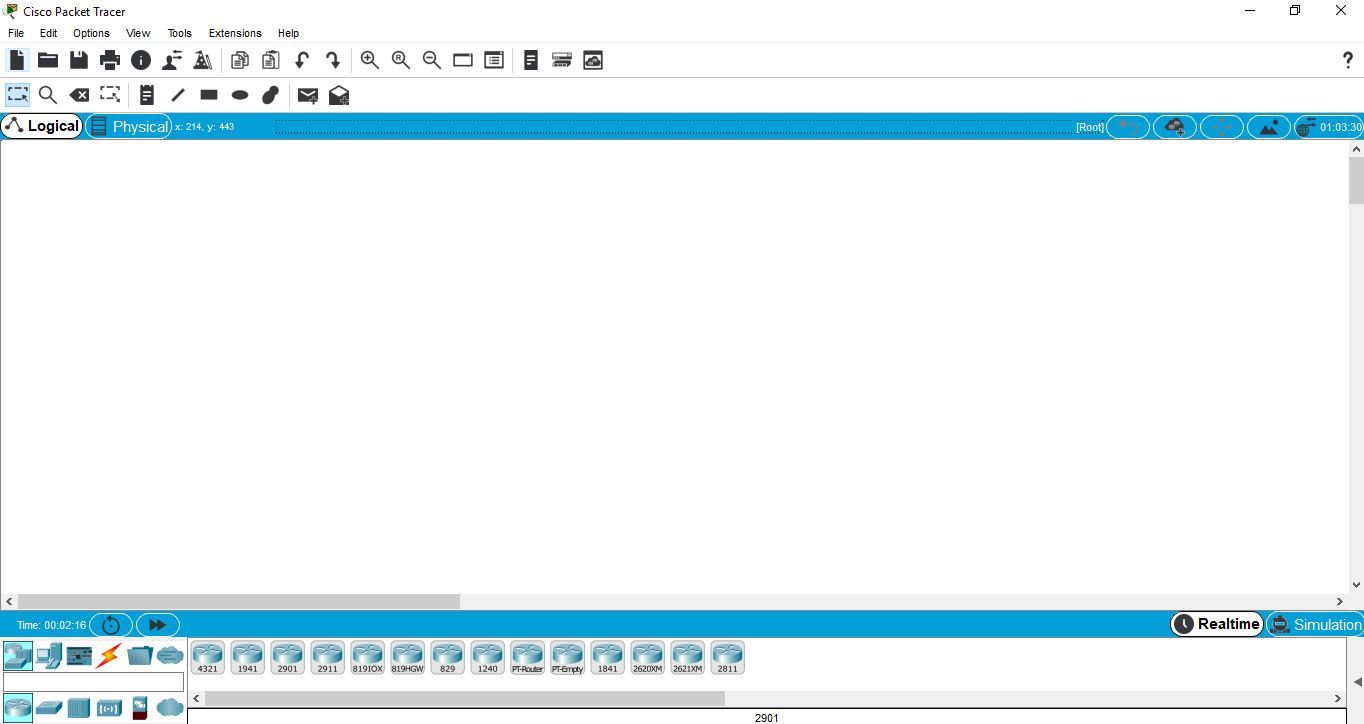
**Characteristics of a Computer Network**

The primary purpose of a computer network is to share resources:

* You may have a computer with a CD writer or a backup system but the other computer doesn’t have it; In this case, you can burn CDs or make backups on a computer that has one of these but using data from a computer that doesn’t have a CD writer or a backup system.
* you can connect a printer (or a scanner, or a fax machine) to one computer and let other computers of the network print (or scan, or fax) to that printer (or scanner, or fax machine)
* You can place a CD with pictures on one computer and let other computers access those pictures
* You can create files and store them in one computer, then access those files from the other computer(s) connected to it.

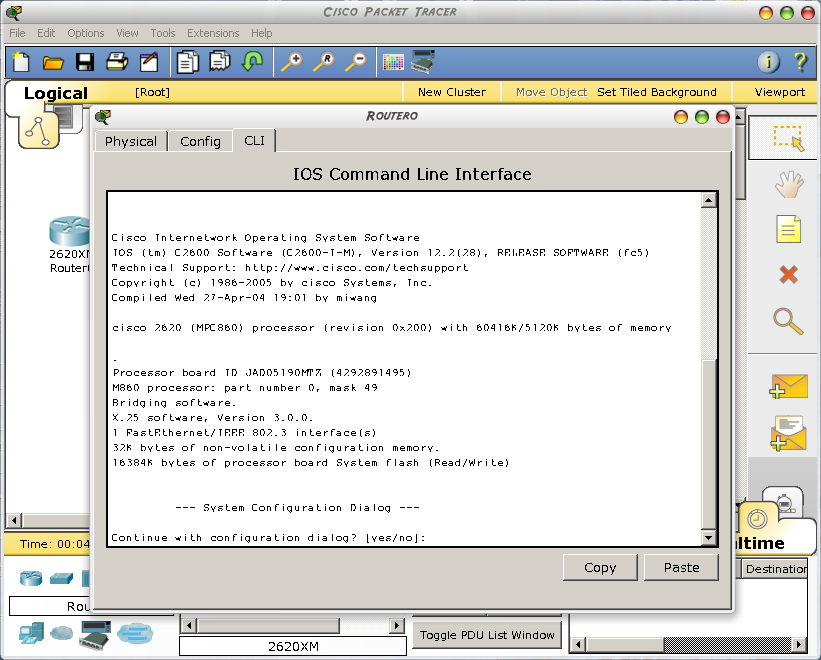
**Introduction to Packet Tracer**

Packet Tracer is a self-paced, visual, interactive teaching and learning tool developed by Cisco. Lab Activities are an important part of networking education. However, lab equipment can be a scarce Resource. Packet Tracer provides a visual simulation of equipment and network processes to offset the challenge of limited equipment. Students can spend as much time as they like completing standard lab. Exercises through Packet Tracer, and have the option to work from home. Although Packet Tracer is not a substitute for real equipment, it allows students to practice using a command-line interface. This “e-doing” capability is a fundamental component of learning how to configure routers and switches from the command line.



**CISCO IOS CLI**

Most Cisco devices (including routers and switches) use a CLI (Command Line Interface) to configure the network device. The CLI is an interface, based on text. You type in configuration commands and use show commands to get the output from the router or switch. There are also GUIs (Graphical User Interface) for the routers, switches and firewalls but the majority of the work is done on the CLI. This might sound dated but with so many commands that are available to use, the CLI is much easier to work with than any of the graphical interfaces. It’s also much easier to copy entire configurations from one device to another. In this lesson, I’ll explain how to access the CLI and the basics of how Cisco IOS works.

Access to Cisco IOS CLI

## CLI EDITOR FEATURES:

The Cisco IOS CLI offers context-sensitive help, a useful tool if you are a new user because at any time during an EXEC session, you can type a question mark ([?](javascript:openCmdRefWindow('?'))) to get help. Two types of context-sensitive help are available: word help and command syntax help.

Command syntax help can be used to obtain a list of command, keyword, or argument options that are available based on the syntax you have already entered. To use command syntax help, enter [?](javascript:openCmdRefWindow('?')) In the place of a keyword or argument.

**Step 1 enable,co ?**

Context-sensitive command syntax help can be used to obtain a list of commands, keywords, or argument options that are available, based on the syntax you have already entered. To use command syntax help to see a list of arguments that can be used with the [configure](javascript:openCmdRefWindow('configure')) command, enter [configure](javascript:openCmdRefWindow('configure')), with ? in the place of a keyword or argument. Include a space before the question mark.

**Step 2: configure ?**

Context-sensitive command syntax help can be used to obtain a list of commands, keywords, or argument options that are available, based on the syntax you have already entered. To use command syntax help to see a list of arguments that can be used with the [configure](javascript:openCmdRefWindow('configure')) command, enter [configure](javascript:openCmdRefWindow('configure')), with ? in the place of a keyword or argument. Include a space before the question mark.

**Step 3:Ctrl-U, li<tab> , Ctrl-U, i<tab> in<tab>**

Lets use another editing feature of the Cisco IOS software. While the characters "in," from the previous lab step, are still being displayed at the prompt, press the Control key, and while holding the Control key down, type a "u." The Control-U key combination, abbreviated "Ctrl-U," erases whatever is displayed on the current, active prompt line in the Cisco IOS software. Control key combinations, like Ctrl-U, are called "hot keys." Continuing with command abbreviation, another way to find the minimum number of characters that uniquely identifies a command is to use the tab-to-complete feature of the Cisco IOS software. For example, type li at the prompt and then press the tab key. The tab-to-complete feature will expand the entry "li" to [line](javascript:openCmdRefWindow('line')), because [line](javascript:openCmdRefWindow('line')) is the only command that begins with "li." Use the Ctrl-U hot key to delete the [line](javascript:openCmdRefWindow('line')) command from the prompt line. Now try typing i at the prompt and then press the tab key. The tab-to-complete feature will not expand this entry, because as we discovered in the previous lab step, i is an ambiguous command. Now enter in at the prompt and then press the tab key. The tab-to-complete feature will expand this entry to [interface](javascript:openCmdRefWindow('interface')), because that is the only command that begins with "in."

**Step 4: Ctrl-R,Ctrl-A,<up-arrow>,<up-arrow>,Ctrl-W,ethernet 0,Ctrl-Z**

You have already learned about the Ctrl-U hot key. The following table offers a comprehensive list of the hot keys and some other keyboard editing functions.

|  |  |
| --- | --- |
| Delete | - Removes one character to the right of the cursor. |
| Backspace | - Removes one character to the left of the cursor. |
| TAB | - Finishes a partial command. |
| Ctrl-A | - Moves the cursor to the beginning of the current line. |
| Ctrl-R | - Redisplays a line. |
| Ctrl-U | - Erases a line. |
| Ctrl-W | - Erases a word. |
| [Ctrl-Z](javascript:openCmdRefWindow('end')) | - Ends configuration mode and returns to the EXEC mode. |
| Up Arrow | - Allows user to scroll forward through former commands. |
| Down Arrow | - Allows user to scroll backward through former commands. |

Use the Ctrl-R hot key to redisplay the current line in the IOS software. Then use the Ctrl-A hot key to move the cursor to the beginning of the current line. Then use the up arrow key on the keyboard to move backwards through the commands in the command buffer—that is, the commands that have been used since entering privileged EXEC mode. Note that a few of the commands you entered in privileged EXEC mode don't appear in the buffer. These are the commands you deleted from the command line with the Ctrl-U hot key; the deletion also removes those commands from the command buffer. You can use the down arrow key to move forward through the buffer. After inspecting the contents of the buffer, use the arrow keys to display "interface serial 11" at the prompt. This was a command with an intentionally invalid parameter, used to demonstrate the command syntax check capability of the Cisco IOS software. Use the Ctrl-W hot key twice, to delete the last two words in this command. Then enter "ethernet 0" after the [interface](javascript:openCmdRefWindow('interface')) command, and go into interface configuration mode for the. Finally, use the Ctrl-Z hot key to return to privileged EXEC mode.

The table below lists commands that can be used to exit out of an operation.

|  |  |
| --- | --- |
| Keystroke | Description |
| Ctrl-C | When in any configuration mode, ends the configuration mode and returns to privileged EXEC mode. |
| Ctrl-Z | When in any configuration mode, ends the configuration mode and returns to privileged EXEC mode. |
| Ctrl-Shift-6 | All-purpose break sequence used to abort DNS lookups, traceroutes, pings, etc. |

**BASIC SWITCH CONFIGURATION**

**Step 1:** [**enable**](javascript:openCmdRefWindow('enable'))**,** [**configure**](javascript:openCmdRefWindow('configure')) **terminal,** [**hostname**](javascript:openCmdRefWindow('hostname')) **Router1**

The default router name is "Switch." Giving each switch a unique name aids in network management by allowing you to uniquely identify each switch within the network. The name of the switch is considered to be the host name and is displayed in the system prompt. Let's change the host name from "Switch" to "Switch1." Notice that the name in the system prompt changes immediately after the [hostname](javascript:openCmdRefWindow('hostname')) command is entered.

|  |  |
| --- | --- |
| **Task** | **Command** |
| 1. From user EXEC mode, enter privileged EXEC mode. | [enable](javascript:openCmdRefWindow('enable')) |
| 1. From privileged EXEC mode, enter global configuration mode. | [configure](javascript:openCmdRefWindow('configure')) terminal |
| 1. From global configuration mode, configure a new host name. | [hostname](javascript:openCmdRefWindow('hostname')) <hostname> |

**Step 2: exit , clock set 11:56:50 11 March 1999,** [**show clock**](javascript:openCmdRefWindow('show%20clock'))

The router will use this information to timestamp error messages. Use the following information for the time and date: 11:56:50 11 March 1999. Then use the [show clock](javascript:openCmdRefWindow('show%20clock')) command to view the new clock setting.

|  |  |
| --- | --- |
| **Task** | **Command** |
| 1. From global configuration mode, return to privileged EXEC mode. | [exit](javascript:openCmdRefWindow('exit')) |
| 1. From privileged EXEC mode, set the system clock. | [clock set](javascript:openCmdRefWindow('clock%20set')) <hh:mm:ss day month year> |
| 1. From privileged EXEC mode, view the clock setting. | [show clock](javascript:openCmdRefWindow('show%20clock')) |

**Step 3:** [**configure**](javascript:openCmdRefWindow('configure')) **terminal,** [**banner motd**](javascript:openCmdRefWindow('banner%20motd')) **# TechnologyDepartment #,** [**exit**](javascript:openCmdRefWindow('exit'))**, <CR>**

The message-of-the-day banner is displayed at login and is useful for sending startup messages to all connected terminals. By default, no message is displayed at logon. Enter the [banner motd](javascript:openCmdRefWindow('banner%20motd')) command in global configuration mode. When using the [banner motd](javascript:openCmdRefWindow('banner%20motd')) command, enter a few blank spaces and then type a delimiting character. The information typed until the next delimiting character is recognized as the message of the day. In this simulation, use the pound key (**#**) as the delimiting character.

**Step 4:** [**enable**](javascript:openCmdRefWindow('enable'))**,** [**configure**](javascript:openCmdRefWindow('configure')) **terminal,** [**interface**](javascript:openCmdRefWindow('interface')) **ethernet 0,** [**description**](javascript:openCmdRefWindow('description')) **FirstEthernetInNetwork1, Ctrl-Z , show running-config**

Now display the running configuration, to see the interface description you just configured. To display the running configuration, use the [show running-config](javascript:openCmdRefWindow('show%20running-config')) command in privileged EXEC mode.

In a configuration display, the exclamation marks function as line separators to make reading easier. In this configuration, notice how commands entered at the interface configuration level, such as the interface description command, appear indented underneath the interface they were entered on. Global-level commands are not indented, so you can easily identify which configuration parameters are set at the global configuration level and which are set at the various configuration sublevels.

Notice the commands that you have configured so far in this Basic Configuration Lab: [hostname](javascript:openCmdRefWindow('hostname')), [service timestamps](javascript:openCmdRefWindow('service%20timestamps')), [banner motd](javascript:openCmdRefWindow('banner%20motd')), and interface [description](javascript:openCmdRefWindow('description')).

|  |  |
| --- | --- |
| **Task** | **Command** |
| 1. From user EXEC mode, enter privileged EXEC mode. | [enable](javascript:openCmdRefWindow('enable')) |
| 1. From privileged EXEC mode, enter global configuration mode. | [configure](javascript:openCmdRefWindow('configure')) terminal |
| 1. From global configuration mode, enter interface configuration mode. | [interface](javascript:openCmdRefWindow('interface')) <type number> |
| 1. From interface configuration mode, configure an interface description. | [description](javascript:openCmdRefWindow('description')) interface\_description |
| 1. From interface configuration mode, return to privileged EXEC mode. | <[Ctrl-Z](javascript:openCmdRefWindow('end'))> |
| 1. From privileged EXEC mode, display the running configuration. | [show running-config](javascript:openCmdRefWindow('show%20running-config')) |

**STEP 5:** [**configure**](javascript:openCmdRefWindow('configure')) **terminal,** [**line console**](javascript:openCmdRefWindow('line')) **0,** [**password**](javascript:openCmdRefWindow('password')) **cisco,** [**Ctrl-Z**](javascript:openCmdRefWindow('end'))**,** [**exit**](javascript:openCmdRefWindow('exit'))**, <CR>, cisco**:

You can use the [password](javascript:openCmdRefWindow('password')) command, in line configuration mode, to configure a password to restrict access to a switch. Console passwords can be established on individual lines. Remember that passwords are case-sensitive. Passwords can be configured for console terminals or for incoming Telnet sessions. Configure the password cisco for console line 0. Then exit the EXEC completely and log back into the switch. You will be prompted for the console password you just configured before you can enter user EXEC mode.

|  |  |
| --- | --- |
| **Task** | **Command** |
| 1. From privileged EXEC mode, enter global configuration mode. | [configure](javascript:openCmdRefWindow('configure')) terminal |
| 1. From global configuration mode, enter line configuration mode for a console line. | [line console](javascript:openCmdRefWindow('line')) <line\_number> |
| 1. From line configuration mode, configure a console password. | [password](javascript:openCmdRefWindow('password'))<password>  Login |
| 1. From line configuration mode, return to privileged EXEC mode. | <[Ctrl-Z](javascript:openCmdRefWindow('end'))> |
| 1. From privileged EXEC mode, exit the EXEC entirely. | [exit](javascript:openCmdRefWindow('exit')) |
| 1. Press Return when prompted. | <CR> |
| 1. Enter the console password at the prompt, to enter user EXEC mode. | password |

**Step 6:** [**enable**](javascript:openCmdRefWindow('enable'))**,** [**configure**](javascript:openCmdRefWindow('configure')) **terminal,** [**line vty**](javascript:openCmdRefWindow('line')) **0 4,** [**password**](javascript:openCmdRefWindow('password')) **cisco**

To further restrict access to the switch, configure the password cisco for zero through four.

|  |  |
| --- | --- |
| **Task** | **Command** |
| 1. From user EXEC mode, enter privileged EXEC mode. | [enable](javascript:openCmdRefWindow('enable')) |
| 1. From Privileged EXEC mode, enter global configuration mode. | [configure](javascript:openCmdRefWindow('configure')) terminal |
| 1. From global configuration mode, enter line configuration mode for a virtual terminal. | [line vty](javascript:openCmdRefWindow('line')) <Starting line\_number><ending\_line\_number> |
| 1. From line configuration mode, configure a virtual-terminal password. | [password](javascript:openCmdRefWindow('password'))<password>  Login |

**Step 7:** [**exit**](javascript:openCmdRefWindow('exit'))**,** [**enable password**](javascript:openCmdRefWindow('enable%20password')) **Cisco,** [**exit**](javascript:openCmdRefWindow('exit'))**,** [**disable**](javascript:openCmdRefWindow('disable'))**,** [**enable**](javascript:openCmdRefWindow('enable'))**, Cisco**

Use the [enable password](javascript:openCmdRefWindow('enable%20password')) global configuration command to configure an enable password to restrict access to privileged EXEC mode. Use Cisco as the password. Then return to user EXEC mode and test the enable password by re-entering privileged EXEC mode.

|  |  |
| --- | --- |
| **Task** | **Command** |
| 1. From line configuration mode, return to global configuration mode. | [exit](javascript:openCmdRefWindow('exit')) |
| 1. From global configuration mode, configure an enable password. | [enable password](javascript:openCmdRefWindow('enable%20password')) **password** |
| 1. From global configuration mode, return to privileged EXEC mode. | [exit](javascript:openCmdRefWindow('exit')) |
| 1. From privileged EXEC mode, return to user EXEC mode. | [disable](javascript:openCmdRefWindow('disable')) |
| 1. From user EXEC mode, reenter privileged EXEC mode, entering the enable password when prompted. | [enable](javascript:openCmdRefWindow('enable'))  password |

**Step 8: configure terminal,** [**logging buffered**](javascript:openCmdRefWindow('logging%20buffered'))

To copy logging messages, such as errors, to an internal buffer, use the [logging buffered](javascript:openCmdRefWindow('logging%20buffered')) command. The buffer is circular, so those newer messages overwrite older ones after the buffer is filled. Having a log of these messages can be useful when troubleshooting network problems. Now let's generate some logging messages.

|  |  |
| --- | --- |
| **Task** | **Command** |
| 1. From privileged EXEC mode, return to global configuration mode. | [configure](javascript:openCmdRefWindow('configure')) terminal |
| 1. From global configuration mode, configure the system to log messages to an internal buffer. | [logging buffered](javascript:openCmdRefWindow('logging%20buffered')) |

**Step 9:** [**Ctrl-Z**](javascript:openCmdRefWindow('end'))**,** [**show interface**](javascript:openCmdRefWindow('show%20interfaces')) **ethernet 0**

To view information about a particular interface, you can use the [show interface](javascript:openCmdRefWindow('show%20interfaces')) command, which provides the following important information:

1. Interface state (for example, UP, DOWN, LOOPED)
2. Protocol addresses
3. Reliability and load
4. Packet rates
5. Error rates
6. Signaling status (that is, From privileged EXEC mode, display the interface information with the [show interface](javascript:openCmdRefWindow('show%20interfaces')) command.

|  |  |
| --- | --- |
| **Task** | **Command** |
| 1. From interface configuration mode, return to privileged EXEC mode. | <[Ctrl-Z](javascript:openCmdRefWindow('end'))> |
| 1. From privileged EXEC mode, display interface information for the interface. | [Show interface](javascript:openCmdRefWindow('show%20interfaces')) <interface\_type> <interface\_number> |

**Step 10: show running-config**

To display the startup configuration that is stored in NVRAM, use the [show startup-config](javascript:openCmdRefWindow('show%20startup-config')) command in privileged EXEC mode.

|  |  |
| --- | --- |
| **Task** | **Command** |
| 1. From privileged EXEC mode, display the startup configuration. | [show startup-config](javascript:openCmdRefWindow('show%20startup-config')) |

**Step 11: show startup-config**

Save the current running configuration by using the [copy running-config startup-config](javascript:openCmdRefWindow('copy')) command to overwrite the start-up configuration. A Cisco IOS switch stores configurations in two locations—RAM and NVRAM. Start-up configuration is used by the switch during operation. Any configuration changes to the switch are made to the running configuration and take effect immediately after the command is entered and is saved in NVRAM and is loaded as the running configuration when the switch boots up. If a switch loses power or is reloaded, changes to the running configuration are lost unless they have been saved to the startup configuration.

Tip: When editing a configuration, save often.

|  |  |
| --- | --- |
| **Task** | **Command** |
| 1. From privileged EXEC mode, save the running configuration. | [copy running-config startup-config](javascript:openCmdRefWindow('copy')) |

Configuration Files

* There are two system files that store the device configuration:
  + - startup-config - This is the saved configuration file that is stored in NVRAM. It contains all the commands that will be used by the device upon startup or reboot. Flash does not lose its contents when the device is powered off.
    - running-config - This is stored in Random Access Memory (RAM). It reflects the current configuration. Modifying a running configuration affects the operation of a Cisco device immediately. RAM is volatile memory. It loses all of its content when the device is powered off or restarted.
    - To save changes made to the running configuration to the startup configuration file, use the copy running-config startup-config privileged EXEC mode command.





Alter the Running Configurations

1. If changes made to the running config do not have the desired effect and the running-config has not yet been saved, you can restore the device to its previous configuration. To do this you can:
2. Remove the changed commands individually.
3. Reload the device using the reload command in privilege EXEC mode. Note: This will cause the device to briefly go offline, leading to network downtime.

**

1. If the undesired changes were saved to the startup-config, it may be necessary to clear all the configurations using the erase startup-config command in privilege EXEC mode.
2. After erasing the startup-config, reload the device to clear the running-config file from RAM.

**Step 12: show version**

Finally, you will use one more command that can be used to collect information about the configuration and status of a switch. This information can be very helpful in troubleshooting switch problems. The [show version](javascript:openCmdRefWindow('show%20version')) command provides much information, including the following:

|  |  |
| --- | --- |
| Software Version | - IOS software version (stored in Flash) |
| Bootstrap Version | - Bootstrap version (stored in boot ROM) |
| System up-time | - Time since last reboot |
| System restart info | - Method of restart (for example, power cycle, crash) |
| Software image name | - IOS filename stored in Flash |
| Router Type and Processor type | - Model number and processor type |
| Memory type and allocation (Shared/Main) | - Main processor RAM  - Shared packet I/O buffering |
| Software Features | - Supported protocols / feature sets |
| Hardware Interfaces | - Interfaces available on router |
| Configuration Register | - Bootup specifications, console speed setting, etc. |

From privileged EXEC mode, display the software version information with the [show version](javascript:openCmdRefWindow('show%20version')) command.

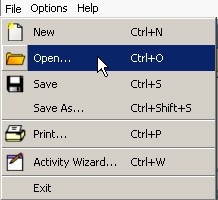
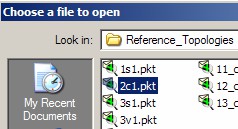
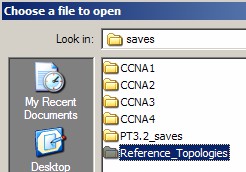
|  |  |
| --- | --- |
| **Task** | **Command** |
| 1. From privileged EXEC mode, display version information. | [show versio](javascript:openCmdRefWindow('show version'))[n](javascript:openCmdRefWindow('show version')) |

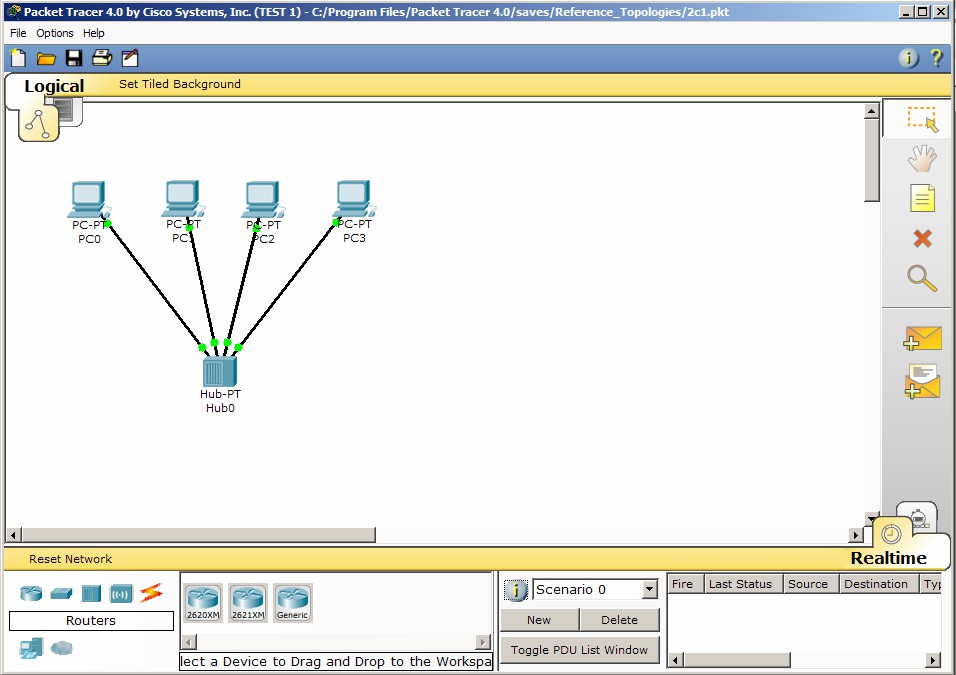
**Overview of Real Time Mode and Simulation Mode**

**Step 1: Start Packet Tracer and Entering Simulation Mode**

**Step 2: Open an existing topology**

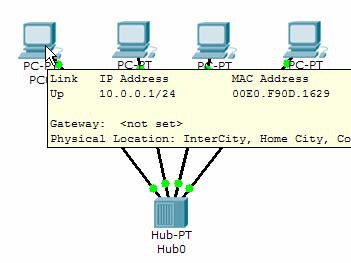
Perform the following steps to open the **2c1.pkt** topology.





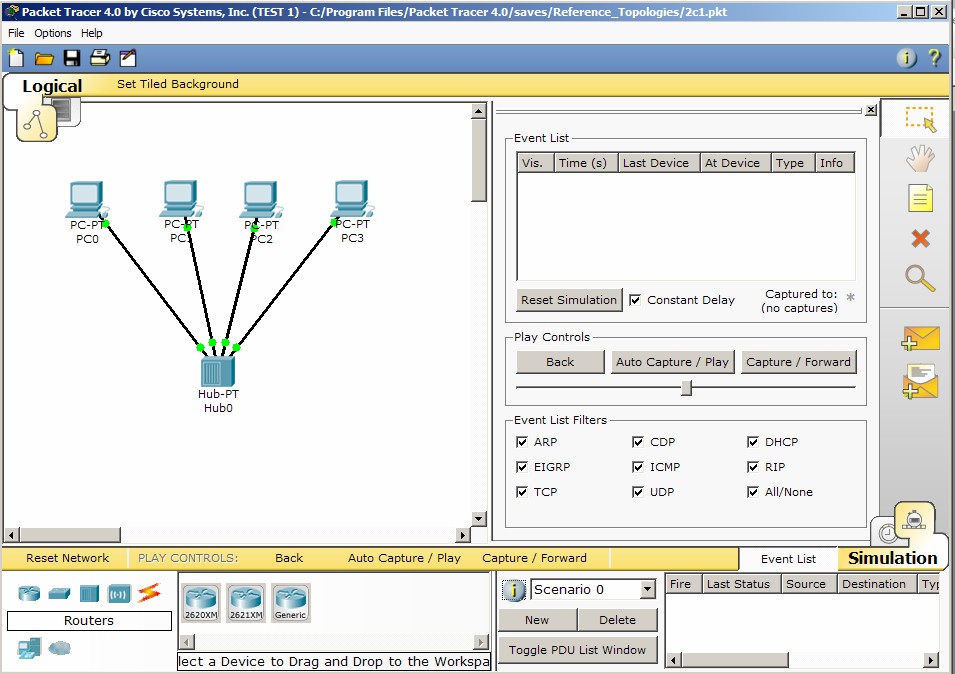
By default, the topology opens in **Realtime** mode. We will examine the difference between **Realtime** and **Simulation** modes in a moment.

**Help** can be obtained by using the Help menu. Both online help one each topic and tutorials are available. Please take advantage of this facilities.





Viewing PC0 information using the **Select** tool:



Once the file is opened, click the **Simulation** icon,

**Step 3: PC0 pinging PC1** :

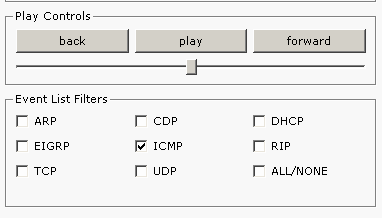
*For those not familiar with ping:* We will examine pings and the ICMP protocol in much more detail later. The ping program generates an IP packet with an encapsulated ICMP Echo Request message. It is a tool used to test basic layer 2 and layer 3 communications between two devices. When the user issues the ping command, most operating systems send multiple (four or five) ICMP Echo messages. When the destination device receives the ping, Echo Request, it issues an Echo Reply.

Command issued from PC0: **ping 192.168.10.37**

Packet Tracer allows us to either issue the command from the command prompt or to use the Add Simple

PDU tool. We will look at both ways to do this.

In order to view only the “pings”, in the **Event List**, click on **ALL/NONE** to clear all protocols, and then click on **ICMP** to select only that protocol.



**2**

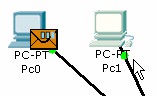
**1**

Using the Simple PDU Tool

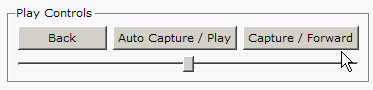
One method for pinging a device from another device is to use the **Simple PDU tool**. This tool performs the ping without having to issue the ping command.

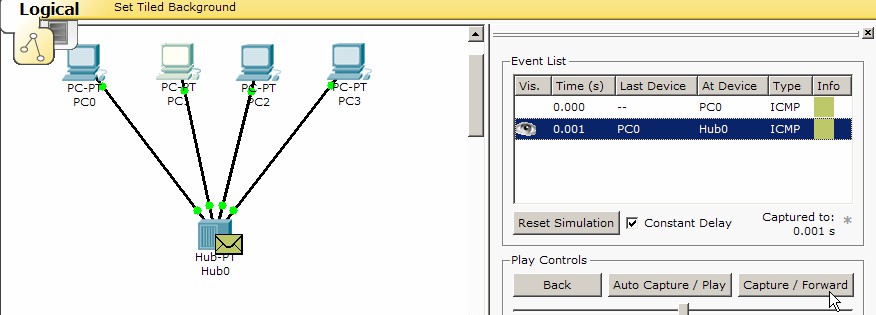
Choose the **Add Simple PDU** tool from the tool box:



Click once on **PC0**, the device issuing the ping (ICMP Echo Request) and then click once on **PC1** (the destination of the ICMP Echo Request).

By clicking on the **Capture/Forward** button, this will forward each event. For example, the first event is the building of the ICMP packet and encapsulating it in an Ethernet frame. The next event will send this Ethernet frame from the Ethernet NIC in PC0 to the Hub.





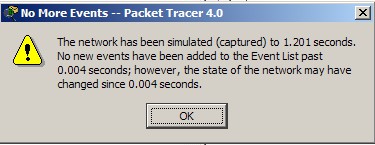
Continue to click on the **Capture/Forward** button and watch the ICMP Echo Requests and ICMP Echo

Replies. ***Notice that the hub floods all of the frames out all ports except the port incoming port.***

Normally, before the ICMP Echo Request, ping, is sent out by PC0, an ARP Request might first be sent. We will discuss this later, but we disabled the display of ARP in the Event List earlier.

**Note**: Using this tool, only a single ping, ICMP Echo Request is sent by PC0, instead of the four pings when using the command prompt.

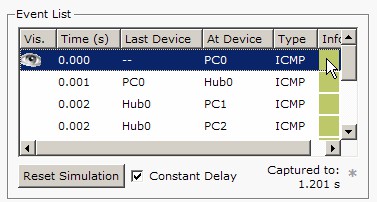
When the ICMP Echo Request and ICMP Echo Reply is finished, you will receive the following message:



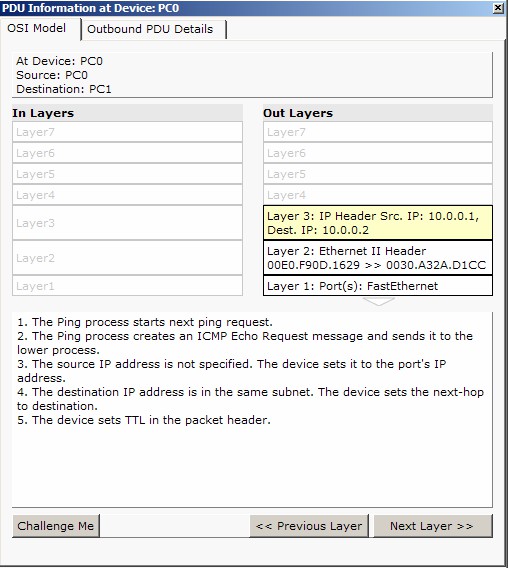
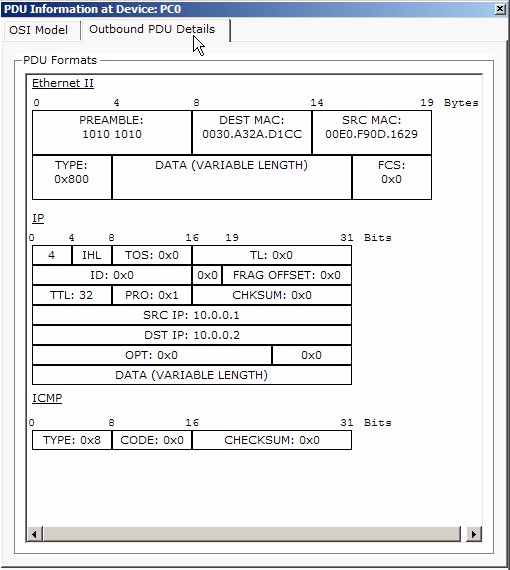
**Step 4: Viewing the frame (Protocol Analyzer)**

To examine the actual protocols being sent, click on the colored **Info** box in the **Event List**. The Event List shows where this Ethernet Frame is currently, “At Device”, the previous devices, “Last Device”, and the type of information encapsulated in the Ethernet Frame, “Info”.

Single click on the first event’s Info box to view the Ethernet frame with the encapsulated IP Packet and the encapsulated ICMP message **“At Device” PC0**.



The PDU (Protocol Data Unit) is displayed in two different formats, **OSI Model** and **Outbound PDU Details**. View them both, paying particular attention to the Layer 2 Ethernet frame. We will discuss IP and ICMP later.



The default is the **OSI Model** view with a brief description with what is occurring with this packet. Click on the **Outbound PDU Details** tab to see the protocol details including the layer 2 Ethernet frame, the layer 3

IP packet and ICMP message.

**HOME ASSIGNMENT**

Q1: Briefly describe any four (4) network types.

Q2: Write description of any three commands of ‘User Mode”?

Q3**:** Write description of any three commands of ‘Privileged Mode”

Q4 : What is the common information listed under the **IP** section of **PDU Details** as compared to the information listed under the **OSI Model** tab? With which layer is it associated?