



Department of CSE
United International University

CSE 324
Computer Networks Lab

Lab Manual

Spring 2018

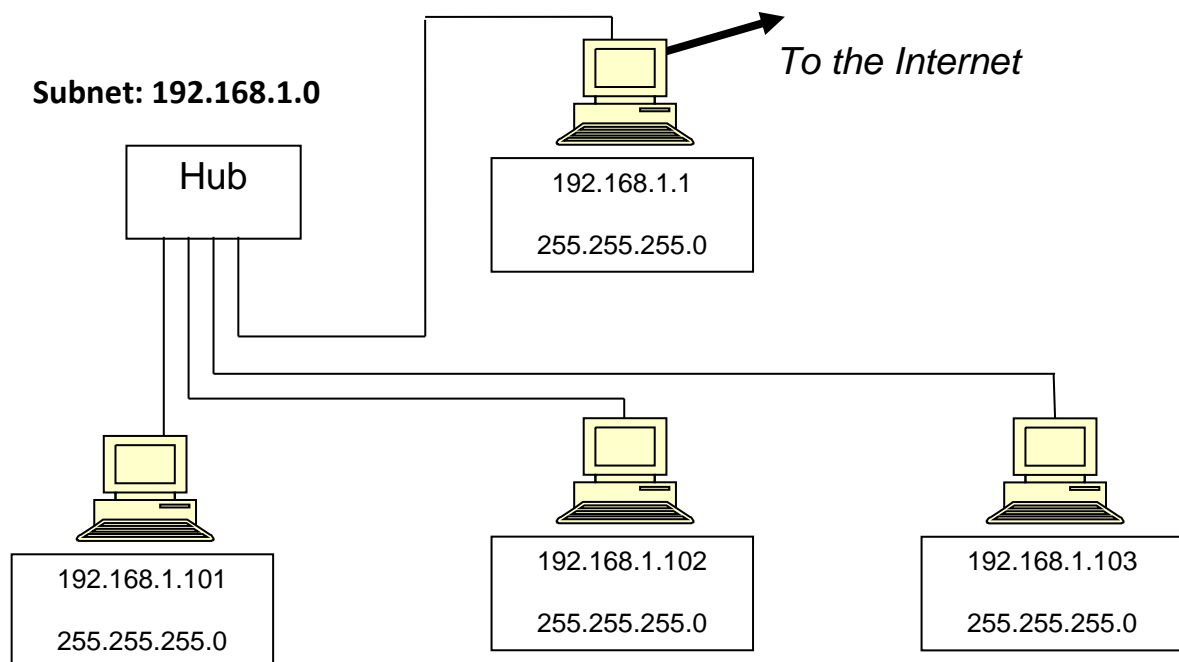
Lab 1 – Part I: Understanding TCP/IP Properties

1. Objectives

- Use of **IP address, Subnet mask, Default gateway, MAC address**.
- Identify tools used to discover a **computer network configuration** in Windows.
- Gather information including **connection, host name, MAC address** and **IP address** information.
- **Compare** network information to other PCs on the network.
- Learn to use the **TCP/IP Packet Internet Groper (ping) command** from a workstation/PC.

2. Background: IP address, Subnet mask, Default gateway, MAC address

Consider the following diagram:

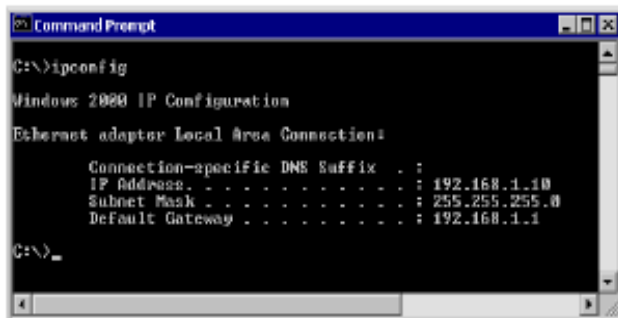


- a. **Subnet Masks:** Subnet mask identifies the network portion of an IP address. Make sure every machine in the same network has the same subnet mask. The value of subnet mask is 255.255.255.0 in the example we will use.
- b. **Network or Subnet Address:** Find the network portion of the IP address of the Gateway Machine. Fill in the host portion with 0s. Write that label above the network (in the upper left, in these diagrams). In the above example, the Gateway Machine has an IP address of 192.168.1.101 and since the subnet mask is 255.255.255.0, the network portion includes only the first 3 bytes. To find the subnet label, replace the last byte with zero: **192.168.1.0**.
- c. **Check the IP Addresses**
Network Portion: Make sure that each NIC on a subnet has the same network address as the label you wrote at the top of the subnet. In the example, on the left subnet, that means every IP address must start with **192.168.1**
Host Portion: Make sure that each NIC on a subnet has a different host address, including the default gateway. In the example, the Gateway Machine has a host address of 1, and the others are 101, 102, and 103, so there are no duplicates.
- d. **Default Gateway:** On each subnet, the default gateway is the Gateway Machine's IP address. It is the same for each NIC on the subnet, except the Gateway Machine itself, which has a default gateway of the network above it, usually an ISP. In the example, the Gateway Machine has an IP address of 192.168.1.1, so the default gateway must be 192.168.1.1 for all three PCs at the bottom of the chart.

3a. Instructions: Experiment 1

Step 1: Gather TCP/IP configuration information

Use the **Start menu** to open the **Command Prompt**, an MS-DOS-like window. Type **ipconfig** and press the **Enter** key. The **ipconfig** is used for gathering the IP Configuration information. The following figure shows the Command screen.



```

C:\>ipconfig

Windows 2000 IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : 
    IP Address. . . . . : 192.168.1.10
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.1.1
  
```

Notes: This first screen shows the **IP address**, **subnet mask**, and **default gateway**. The **IP address** and the **default gateway** should be in the **same network** or **subnet**, otherwise this host would not be able to communicate outside the network. In the figure the **subnet mask** tells us that the **first three octets** must be the same to be in the same network.

Step 2: Record the following TCP/IP information for this computer

IP address: _____

Subnet Mask: _____

Default Gateway: _____

Step 3: Check additional TCP/IP configuration information

To see detailed information, type **ipconfig /all** and press **Enter**. The figure shows the detailed IP config. screen. The **host name**, including the computer name should be displayed. Notice the **Physical Address (MAC)** and the **NIC (Network Interface Card) model (Description)**. All machines share the **first three Hex pairs** in the adapter address. These three pairs identify the **manufacturer of the adapter**.



```

C:\>ipconfig /all

Windows 2000 IP Configuration

Host Name . . . . . : clouds
Primary DNS Suffix . : 
Node Type . . . . . : Hybrid
IP Routing Enabled. . . . . : No
WINS Proxy Enabled. . . . . : No

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : 
    Description . . . . . : LNE100TX Fast Ethernet Adapter Version 1.0
    Physical Address. . . . . : 00-80-C2-23-1E-40
    DHCP Enabled. . . . . : Yes
    Autoconfiguration Enabled . . . . : Yes
    IP Address. . . . . : 192.168.1.10
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.1.1
    DHCP Server . . . . . : 192.168.1.1
    DNS Servers . . . . . : 24.0.124.34
                             24.0.124.34
    Lease Obtained. . . . . : Tuesday, January 07, 2003 10:56:19 AM
    Lease Expires . . . . . : Monday, January 15, 2003 10:56:19 AM
  
```

Write down the following from the output:

a) **HostName**: _____

b) IP addresses of **DNS server**: _____

Do your PC and DNS Server above **share the same network portion?** (yes/No)

If some or all of the servers and PCs is in another network, it means that the **default gateway** is going to **forward** requests to the other network.

Step 4: Compare the TCP/IP configuration of this computer to another computer on the LAN

Select another **computer B**. Perform **step 2 & 3** for the computer B. **Fill up** the following table.

	Your Computer	Computer B	Are there any similarities? (yes/no)
IP Address:			
Subnet Mask:			
Default Gateway:			
MAC Address:			

The IP addresses should share the **same network portion**. All PCs in the LAN should share the **same default gateway**.

Step 5: Close the screen

Close the screen when finished examining network settings.

3b. Instructions: Experiment 2

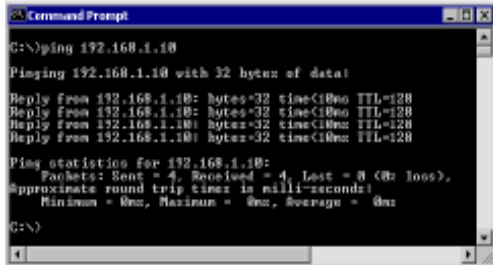
You need the **IP addresses** that were recorded in **experiment 1**.

Step 1: Access the command prompt

Use the Start menu to open the Command Prompt window.

Step 2: ping the IP address of another computer

In the window, type **ping**, a space, and the **IP address of another computer B**. The following figure shows the successful results of **ping** to this IP address.



```
G:\>ping 192.168.1.10
Pinging 192.168.1.10 with 32 bytes of data:
Reply from 192.168.1.10: bytes=32 time<10ms TTL=128
Reply from 192.168.1.10: bytes=32 time<10ms TTL=128
Reply from 192.168.1.10: bytes=32 time<10ms TTL=128
Reply from 192.168.1.10: bytes=32 time<10ms TTL=128

Ping statistics for 192.168.1.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
G:\>
```

Ping uses the **ICMP echo request** and **echo reply** feature to **test physical connectivity**. Since **ping** reports on **four attempts**, it gives an indication of the reliability of the connection. Now, **ping** the IP address of **Computer B**. Look over the results and verify that the **ping** was successful. Is the **ping** successful? (yes/no) _____.

Note the results: **Packets: sent = __, Received = __, Lost = __**

Step 3: ping the IP address of the default gateway

Try to **ping** the **IP address of the default gateway** listed in the last exercise. If the **ping** is successful, it means there is physical connectivity to the router on the local network and probably the rest of the world. Was the **ping** successful? (yes/no) _____

Step 4: ping the Loopback IP address of this computer

Type the following command: **ping 127.0.0.1** The 127.0.0.0 network is reserved for loopback testing. If the **ping** is successful, then TCP/IP is properly installed and functioning on this computer. Was the **ping** successful? (yes/no) _____

Demonstrate your work to the instructors before leaving.

Signature of the Instructor

Lab 1 – Part II: Introduction to Packet Tracer, Building a single-segment Network

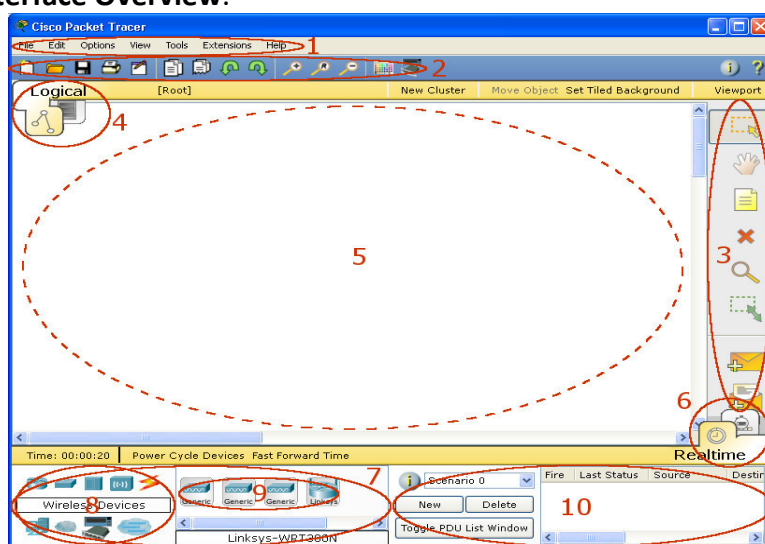
1. Objectives

- To introduce **Packet Tracer (PT)** and become familiar with its operations.
- Create a simple network using **Hub, Switch**.

2a. Instructions

Packet Tracer (PT) is a network simulator that enables you build, configure, observe, initiate, modify and troubleshoot networks and network activity. It allows you to observe and better understand how data (packets) travel across a network, as well as enabling you to configure routers and switches using Cisco's IOS (Internetwork Operating System).

Exercise 1: Open Packet Tracer (**start -> cisco packet tracer**) and click on **help -> Contents**. Go to: **Getting Started -> Interface Overview**.



Familiarize yourself with the functions of the two screen modes; **Realtime** and **Simulation**.

- **Real time Mode** is used to build and configure your network,
- **Simulation mode** is used to generate network traffic (packets) and observe network activity.

2b. Crossover and Straight-through Cables

Common Ethernet network cables are **straight** and **crossover cable**. This Ethernet network cable is made of **4 pair high performance cable** that consists of **twisted pair conductors** that used for data transmission. Both end of cable is called **RJ45 connector**. Straight and crossover cable can be Cat3, Cat 5, Cat 5e or Cat 6 UTP cable, the only difference is each type will have different wire arrangement in the cable for serving different purposes.



Category 1 (Layer 2 or below)

Hub
Switch

Category 2 (Layer 3 or above)

PC/Laptop/Server
Router

You usually use **straight cable** to connect **different type of devices (different category)**. This type of cable will be used most of the time and can be used to:

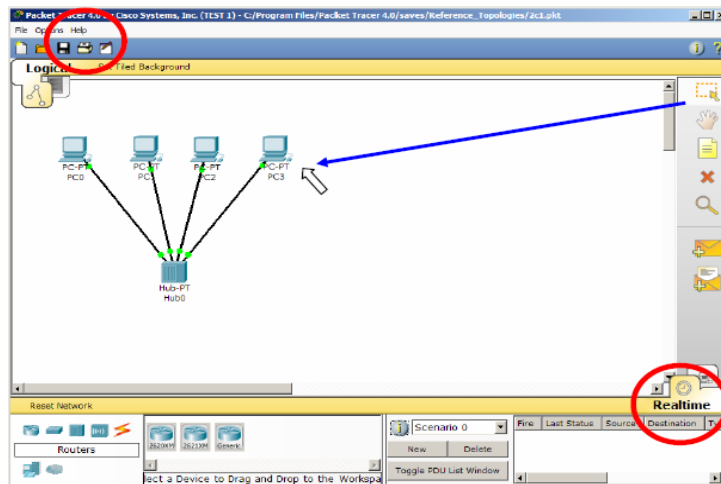
- ✓ Connect a computer to a switch/hub's normal port.
- ✓ Connect a computer to a cable/DSL modem's LAN port.
- ✓ Connect a router's WAN port to a cable/DSL modem's LAN port.
- ✓ Connect a router's LAN port to a switch/hub's uplink port. (normally used for expanding network)

Sometimes you will use **crossover cable**, it's usually used to **connect same type of devices (same category)**. A crossover cable can be used to:

- ✓ Connect 2 computers directly.
- ✓ Connect a router's LAN port to a switch/hub's normal port. (normally used for expanding network)
- ✓ Connect 2 switches/hubs by using normal port in both switches/hubs.

Exercise 2: To create a **single-segment network** using a Hub and a Switch.

Step 1: Start Packet Tracer and create the shown topology using Hub, Switch and PCs.



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

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

To view the **IP address, subnet mask, default gateway, and MAC address of a host**, move the cursor over that computer.

Step 2: Configure the PCs with **Host IP Address, Subnet Mask** as follows:

Host	IP Address	Subnet Mask
PC0	192.168.10.10	255.255.255.0
PC1	192.168.10.11	255.255.255.0
PC2	192.168.10.12	255.255.255.0
PC3	192.168.10.13	255.255.255.0

Step 3: Test the network, e.g., ping PC3 from PC0.

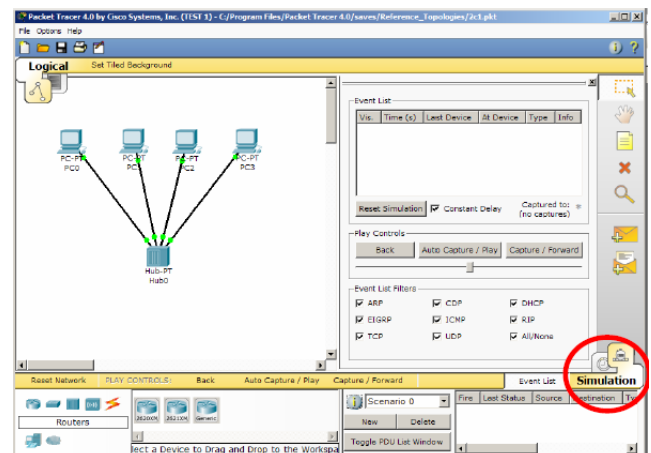
Step 4: Run the test in **Simulation mode**.

Once the file is opened, click the **Simulation** icon, to enter **simulation mode**. Simulation mode allows you to view the **sequence of events** associated with the communications between two or more devices.

Realtime mode performs the operation with all of the sequence of events happening at "real time".

1. Set **IP address, SM and Default gateway** in a PC
2. Ping from the command prompt
3. Use the web browser of a PC
4. Difference of **real-time** and **simulation modes** using a simple PDU.

Step 5: Change the IP address of PC3 to **192.168.20.13**. Perform a **ping from PC0 to PC3**. What is the ping result?



Step 6: Return the IP address of PC3 to **192.168.10.13**. Change the IP address of PC2 to **192.168.11.12**. Perform a ping from PC0 to PC2. What is the ping result?

Exercise 3: From the all Exercises you have done, observe and explain at least **4 points** about hub vs. switch and real-time vs. simulation modes.

Point 1:	
Point 2:	
Point 3:	
Point 4:	

Demonstrate your work to the instructors before leaving.

Signature of the Instructor

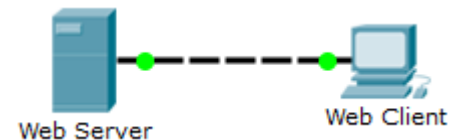
Lab 2 – Part I: Protocol analysis using OSI/TCP-IP layering models, Identify IP and MAC Address

1. Objectives

- To explore the **OSI layering model** using **Packet Tracer**.
- To identify **IP** and **MAC addresses**.

2. Instructions

This simulation activity is intended to provide a foundation for understanding the TCP/IP protocol suite and the relationship to the OSI model. Simulation mode allows you to view the data contents being sent across the network at each layer. Create the following topology:



Part 1: Examine HTTP Web Traffic

Step 1: Switch from Real time to Simulation mode.

- a. Click the **Simulation** mode icon to switch from **Realtime** mode to **Simulation** mode.
- b. Select **HTTP** from the **Event List Filters**.

Step 2: Generate web (HTTP) traffic.

- a. Click **Web Client** in the far left pane.
- b. Click the **Desktop** tab and click the **Web Browser** icon to open it.
- c. In the URL field, enter www.osi.local and click **Go**.
- d. Click **Capture/Forward** four times. There should be four events in the Event List.

Step 3: Explore the contents of the HTTP packet.

- a. Click the first colored square box under the **Event List > Info** column.
- b. Ensure that the **OSI Model** tab is selected. Under the **Out Layers** column, ensure that the **Layer 7** box is highlighted. What is the text displayed next to the **Layer 7** label? _____
- c. Click **Next Layer**. Layer 4 should be highlighted. What is the **Dst Port** value? _____
- d. Click **Next Layer**. Layer 3 should be highlighted. What is the **Dest. IP** value? _____
- e. Click **Next Layer**. What information is displayed at this layer? _____
- f. Click the **Outbound PDU Details** tab.

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?

What is the common information listed under the **TCP** section of **PDU Details**, as compared to the information listed under the **OSI Model** tab, and with which layer is it associated?

What is the **Host** listed under the **HTTP** section of the **PDU Details**? What layer would this information be associated with under the **OSI Model** tab?

- g. Click the **next colored square box** under the **Event List > Info** column. Only Layer 1 is active (not grayed out). The device is **moving the frame from the buffer and placing it on to the network**.
- h. Advance to the next HTTP **Info** box within the **Event List** and click the colored square box. This window contains both **In Layers** and **Out Layers**. **The server is now sending the information back to the client.**

Comparing the information displayed in the **In Layers** column with that of the **Out Layers** column, what are the **major differences**?

- i. Click the **Outbound PDU Details** tab. Scroll down to the **HTTP** section. What is the first line in the **HTTP message** that displays?
 - j. Click the **last colored square box** under the **Info** column. How many tabs are displayed with this event and **why**?
-

Part 2: Display Elements of the TCP/IP Protocol Suite

In Part 2 of this activity, you will use the Packet Tracer **Simulation mode** to view and examine some of the other protocols comprising of the **TCP/IP suite**.

Step 1: View Additional Events

- a. Close any open PDU information windows.
- b. In the **Event List Filters > Visible Events** section, click **Show All**. What additional **Event Types** are displayed?
- c. Click the **first DNS event** in the **Info** column. As you look at the **OSI Model** tab with **Layer 7** highlighted, a description of what is occurring is listed directly below the **In Layers** and **Out Layers** ("1. **The DNS client sends a DNS query to the DNS server.**"). This is very useful information to help understand what is occurring during the communication process.
- d. Click the **Outbound PDU Details** tab. What information is listed in the **NAME:** in the DNS QUERY section?
- e. Click the last DNS **Info** colored square box in the event list. Which device is displayed?

What is the value listed next to **ADDRESS:** in the DNS ANSWER section of the **Inbound PDU Details**?

Demonstrate your work to the instructors before leaving.

Signature of the Instructor

Lab 2 – Part II: IP Addressing and Subnetting

1. Objectives

- To introduce **IPv4 addressing** and **subnetting**.
- To design an **enterprise network IP addressing scheme**.

2. Instructions

- a) Use [**Handout_1_Basic_Subnetting_Algorithm**] for introduction to IP addressing and subnetting.
- b) Understand the following **IPv4 table**:

Address Class	1st octet range (decimal)	1st octet bits (green bits do not change)	Network(N) and Host(H) parts of address	Default subnet mask (decimal and binary)	Number of possible networks and hosts per network
A	1-127**	00000000-01111111	N.H.H.H	255.0.0.0	128 nets (2^7) 16,777,214 hosts per net (2^{24-2})
B	128-191	10000000-10111111	N.N.H.H	255.255.0.0	16,384 nets (2^{14}) 65,534 hosts per net (2^{16-2})
C	192-223	11000000-11011111	N.N.N.H	255.255.255.0	2,097,150 nets (2^{21}) 254 hosts per net (2^{8-2})

Now, use the above IPv4 address table, **determine** the **class**, **network address** and **broadcast address** for the following IP addresses:

- i. IP address: **207.21.54.240** Address Class: _____
 Subnet mask: **255.255.255.0** Network Address: _____
 Broadcast Address: _____ Possible # of Hosts: _____
- ii. IP address: **60.41.211.5** Address Class: _____
 Subnet mask: **255.0.0.0** Network Address: _____
 Broadcast Address: _____ Possible # of Hosts: _____
- iii. IP address: **190.101.2.199** Address Class: _____
 Subnet mask: **255.255.0.0** Network Address: _____
 Broadcast Address: _____ Possible # of Hosts: _____

c) Determine the **class and major network address** for the following IPv4 addresses. Use the **subnet masks** to determine the **number of subnets created for the address**, and the **number of hosts** permitted on each subnet:

- i. IP address: **207.21.54.140** Address Class: _____
 Subnet mask: **255.255.255.224** Network Address: _____
 Possible # of Subnets: _____ Possible # of Hosts: _____
- ii. IP address: **60.41.211.5** Address Class: _____
 Subnet mask: **255.255.255.0** Network Address: _____
 Possible # of Subnets: _____
 Possible # of Hosts: _____
- iii. IP address: **182.191.25.11** Address Class: _____
 Subnet mask: **255.255.254.0** Network Address: _____
 Possible # of Subnets: _____ Possible # of Hosts: _____

Demonstrate your work to the instructors before leaving.

Signature of the Instructor

Lab 3: Basic Router Configuration, IP Addressing and Subnetting (contd.)

1. Objectives

- Learn about **basic components of a router**.
- Configure **router interfaces** and learn other **basic router configuration settings**.
- To connect hosts in different networks using a Router in **Packet Tracer**.

2. Theoretical background

Part 1: CISCO Internet Operating System (IOS) Command Interface User Levels

The following table contains the different IOS command modes, their roles and the shape of the command prompt that illustrates the mode. Make sure to study this table carefully as it is essential for proper working with Cisco routers and switches.

IOS command mode	Functions/Roles of the mode	Command prompt
User EXEC mode	<ul style="list-style-type: none"> ✓ Limited command set, e.g., ping, telnet, traceroute ✓ No change of system parameters 	Router>enable
Privileged EXEC mode	<ul style="list-style-type: none"> ✓ Manage configuration files ✓ Examine state of router ✓ Access control with password 	Router#configuration terminal
Global configuration mode	<ul style="list-style-type: none"> ✓ Change system wide configuration parameters 	Router(config)#interface fa0/0
Other configuration mode	<ul style="list-style-type: none"> ✓ Interface configuration mode: Modify configuration of a specific interface ✓ Router configuration mode: Modify configuration of specific routing protocol 	Router(config-if)# Router(config-router)#
✓ Use TAB and "?" mark frequently for command		

a) User Exec Mode

- ✓ The user EXEC mode is entered when the router is accessed via a serial connection or when accessing the router via telnet.
- ✓ The command prompt of the user EXEC mode is: **Router>**
- ✓ The user EXEC mode only offers a small set of commands, such as ping, telnet, and traceroute.
- ✓ Configuration parameters cannot be read or modified in this mode.
- ✓ Logging the user off, type: **Router> exit**

b) Privileged EXEC Mode

- ✓ To change or view configuration information of a router, user must enter system administrator mode called Privileged EXEC Mode
- ✓ The privileged EXEC mode is used to read configuration files, reboot the router, and set operating parameters.
- ✓ Entering the privileged EXEC mode requires to type a password, called the **enable secret**.
- ✓ The privileged EXEC mode is entered by this command: **Router>enable**
- ✓ If a password is set, then the system will require it at this stage. Typing the password displays the following command prompt: **Router1#**
- ✓ For logging off, type: **Router1#disable**

c) Global Configuration Mode

- ✓ The global configuration mode is used to modify system wide configuration parameters, such as routing algorithms and routing tables.
- ✓ This is done by typing:

Router1#Configure terminal

- ✓ The argument terminal tells the router that the configuration commands will be entered from a terminal. The alternatives are to issue configuration commands from a configuration file or from a remote machine via a file transfer.
- ✓ The command prompt in the global configuration mode is:

Router1(Config)#

Notes:

- ✓ Typing a question mark (?) in a given command mode generates a list of all available commands in the current command mode

Router1(config-if)#?

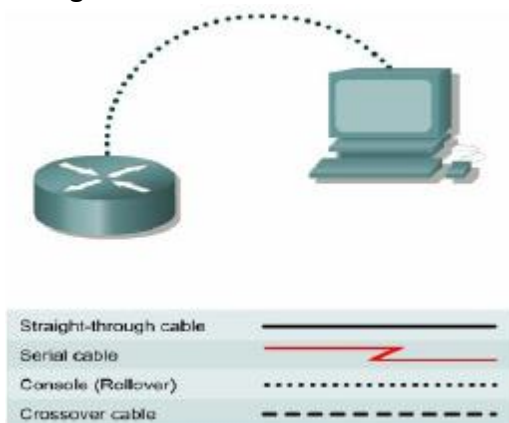
- ✓ This command helps to determine if a command can be executed in the current mode
- ✓ The question mark can also be used to determine the list of available options of a command.

Router1#configure ?

- ✓ If a certain command enables a feature of a router than adding a “no” in front of that command disables the same feature.
- ✓ Disable a network interface: *shutdown*
- ✓ Enable a network interface : *no shutdown*

Part 1.1: Configure PT Terminal to Establish a Console Session with a Cisco IOS Router/Switch.

PT Terminal is a simple emulation program for serial communication that can be used to connect to the **console port** on Cisco IOS devices. A serial interface on a computer is connected to the Cisco device via a **console cable**. Using PT Terminal is the most basic way to access a router for checking or changing its configuration.

**Steps:**

- 1- Select a **PC** and a **router** from the Network Component Box.
- 2- Connect the **console (rollover) cable** to the **console port** on the router. Connect the other cable end to the host computer with a **RS-232 port**.
- 3- From the Windows taskbar, start the PT Terminal program by clicking **PC0>Desktop Tab> Terminal**
- 4- Click OK.
- 5- You should see a response from the router on the screen (press enter several times).

Note: this process is the same for a **Switch**.

Part 1.2: Establishing a Console Session with HyperTerminal

HyperTerminal is a simple Windows-based terminal emulation program for serial communication that can be used to connect to the console port on Cisco IOS devices. A **serial interface** (RS-232) on a computer is connected to the Cisco device via a **rollover cable**. Using HyperTerminal is the most basic way to access a router for checking or changing its configuration.

3. Creating a Local Area Network (LAN) using Hub/Switch/Router

3a. Instructions: Using a Router to connect two different groups/subnets

Task 1: Physical Connections

Create **2 groups** (subnets – network address will be different) of PCs and connect all PCs in a group with hub/switch. Then connect 2 groups/subnets with a **router**.

Task 2: Configure a Router

Step 1: Use the privileged mode and configuration modes.

There are **four (4) IOS modes**: 1) User, 2) Privileged, 3) Global Configuration, and 4) Interface mode.

- ✓ To enter privileged mode, use command: **enable** from user mode.
- ✓ To enter **global configuration mode**, enter command: **configuration terminal (config t)** at privileged mode.
- ✓ There are commands that may be used to exit the current configuration mode: **exit** (to go one step up) and **end** (to go to **privileged mode** directly).

Task 3: Configure the Router Interfaces

Write down your IP address and mask of first network (Fa0/0):

Write down your IP address and mask of first network (Fa0/1):

Step 1: Configure the router fa0/0 interface.

```
Router1(config)# interface fa0/0
Router1(config-if)# description Connection to Host1 with crossover cable
Router1(config-if)# ip address address mask
Router1(config-if)# no shutdown
Router1(config-if)# end
Router1#
```

Step 2: Configure the router fa0/1 interface.

```
Router1(config)# interface fa0/1
Router1(config-if)# description Connect to switch with straight-through cable
Router1(config-if)# ip address address mask
Router1(config-if)# no shutdown
Router1(config-if)# end
Router1#
```

[**Note:** Observe how the router interface state changes after giving the “no shutdown” command.]

Step 3: Configure the host computers.

Configure the host computers for LAN. Fill in the following fields for **network 1 (connected with Fa0/0)**:

IP Address: The first host address _____

Subnet Mask: _____

Default Gateway: Router's IP Address _____

Fill in the following fields for **network 2 (connected with Fa0/1)**:

IP Address: The first host address _____

Subnet Mask: _____

IP Address: The second host address _____

Subnet Mask: _____

Default Gateway: Router's IP Address _____

Step 4: Verify network connectivity.

Use the **ping** command to verify network connectivity with the router. If ping replies are not successful troubleshoot the connection:

Task 4: Save the Router Configuration File.

Cisco IOS refers to RAM configuration storage as **running-configuration**, and NVRAM configuration storage as **startup-configuration**. For configurations to survive rebooting or power restarts, the RAM configuration must be copied into **non-volatile RAM (NVRAM)**. This does not occur automatically, NVRAM must be manually updated after any changes are made.

Step 1: Compare router RAM and NVRAM configurations.

Use the Cisco IOS **show** command to view RAM and NVRAM configurations. The configuration is displayed one screen at a time. A line containing “-- **more** --” indicates that there is additional information to display. The following list describes acceptable key responses:

- ✓ Display the contents of NVRAM. If no output for NVRAM is shown, it is because there is no saved configuration:
Router1# **show startup-config**
startup-config is not present
Router1#
- ✓ Display the contents of RAM.
Router1#**show running-config**

Step 2: Save RAM configuration to NVRAM.

For a configuration to be used the next time the router is powered on or reloaded, it must be manually saved in NVRAM. Save the RAM configuration to NVRAM:

```
Router1# copy running-config startup-config
Destination filename [startup-config]? <ENTER>
Building configuration...
[OK]
```

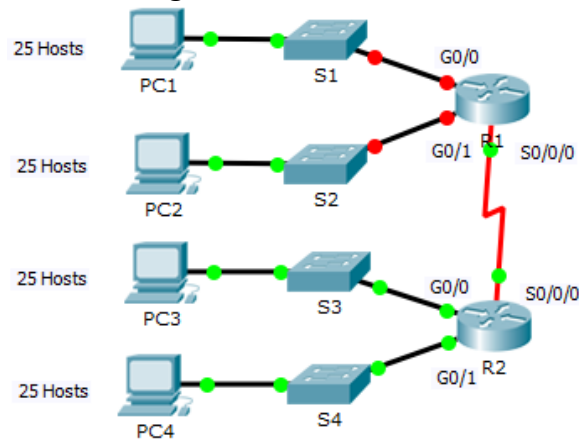
Answer the following:

1. What are the **four primitive modes** of Router IOS? What are the **purposes of each mode**?
2. Why we use **TAB** and **?** symbol during router configuration?
3. What is the **basic difference** between “**startup configuration**” file and “**running configuration**” file?
4. What is the output of “**show run**” command?
5. **Why** and in **which mode** “**copy run start**” command is used?
6. **Why** “**no shutdown**” command is used?

Demonstrate your work to the instructors before leaving.

Signature of the Instructor

4. Exercise – IP Addressing and Subnetting



Step 1: Subnet the **192.168.100.0/24** network into the **appropriate number of subnets**.

- Based on the topology, **how many subnets** are needed?.....
- How many bits** must be borrowed to support the number of subnets in the topology table?.....
- How many subnets** does this create?.....
- How many usable hosts** does this create per subnet?.....
- Calculate the binary value for the **first five subnets**. The first subnet is already shown.
 Net 0: 192 . 168 . 100 . 0 0 0 0 0 0 0 0
 Net 1: 192 . 168 . 100 . _____
 Net 2: 192 . 168 . 100 . _____
 Net 3: 192 . 168 . 100 . _____
 Net 4: 192 . 168 . 100 . _____
- Calculate the binary and decimal value of the new subnet mask.
 11111111.11111111.11111111. _____
 255 . 255 . 255 . _____
- Fill in the **Subnet Table**, listing the decimal value of all available subnets, the first and last usable host address, and the broadcast address. Repeat until all addresses are listed.

Subnet Table

Subnet Number	Subnet Address	First Usable Host Address	Last Usable Host Address	Broadcast Address
0				
1				
2				
3				
4				
5				
6				
7				

Step 2: Assign the subnets to the network shown in the topology.

- Assign Subnet 0 to the LAN connected to the GigabitEthernet 0/0 interface of R1:.....
- Assign Subnet 1 to the LAN connected to the GigabitEthernet 0/1 interface of R1:.....
- Assign Subnet 2 to the LAN connected to the GigabitEthernet 0/0 interface of R2:.....
- Assign Subnet 3 to the LAN connected to the GigabitEthernet 0/1 interface of R2:.....
- Assign Subnet 4 to the WAN link between R1 to R2:.....

Step 3: Document the addressing scheme.

- a. Assign the **first usable IP addresses** to R1 for the two LAN links and the WAN link.
- b. Assign the **first usable IP addresses** to R2 for the LANs links. Assign the **last usable IP address** for the WAN link.
- c. Assign the **second usable IP addresses** to the switches.
- d. Assign the **last usable IP addresses** to the hosts.

Step 4: Assign IP Addresses to Network Devices and Verify Connectivity

Most of the IP addressing is already configured on this network. Implement the following steps to complete the addressing configuration.

- a. Configure IP addressing on R1 LAN interfaces.
- b. Configure IP addressing on S3, including the default gateway.
- c. Configure IP addressing on PC4, including the default gateway.
- d. Verify connectivity.

[IP Addressing and Subnetting – More practice]

Q.1 You have the IP address **186.111.0.0**, this network is subnetted by **10-bits**. Find the following:

- i. Find the **Subnet Mask**.
- ii. Determine the **number of usable hosts per subnet**.
- iii. To **which subnet** the following IP's belong to: **186.111.169.213**
- iv. Determine the **network address** and **broadcast address** of the subnet to which this ip belongs to: **186.111.169.213**
- v. Find **Network Address, Broadcast Address** and **Host Range** for the **subnet # 121**.

Q.2 Given a host with IP address **160.50.145.189/21**:

- i. Is a host with IP address **160.50.146.210/21** part of the same network? **Show calculations**.
- ii. Is the IP address **160.50.145.255** valid according to the given IP? Why or why not?
- iii. What is the **first valid host** on the subnetwork that the node **172.18.142.179 255.255.254.0** belongs to?
- iv. **Which subnet** does host **192.168.11.198 255.255.255.240** belong to?
- v. What is the **last valid host** on the subnetwork **192.168.98.176 255.255.255.240**?
- vi. What is the **last valid host** on the subnetwork **172.25.13.112 255.255.255.240**?
- vii. **How many subnets** and hosts per subnet can you get from the network **10.0.0.0 255.255.240.0**?
- viii. What is the **first valid host** on the subnetwork that the node **192.168.207.190/28** belongs to?

Demonstrate your work to the instructors before leaving.

Signature of the Instructor

Lab 4: Internetworking using Routers/Routing Protocols (Static, Default and Dynamic Routing)

1. Objectives

- To learn how to connect and configure more than one **Router**
- Using **static and default routes**
- Using a **dynamic routing protocol (RIP)**
- To build and configure an **internetwork** using **Packet Tracer**

2. Background: Static and Default Routes

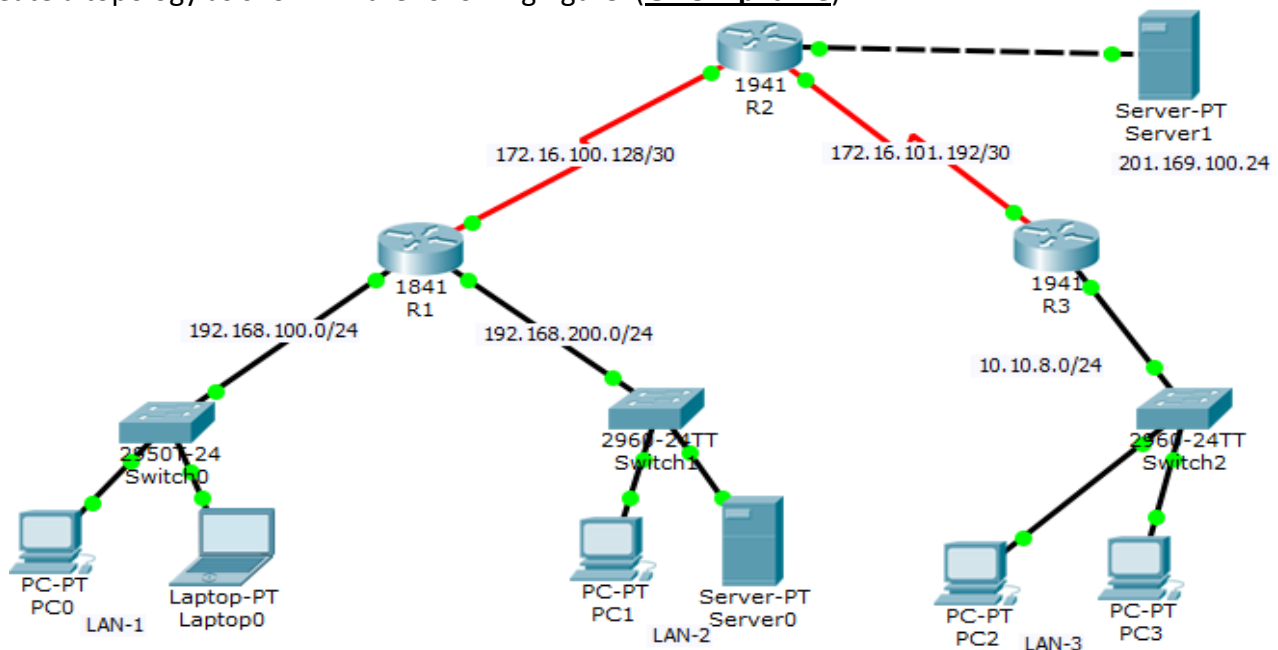
In this activity, you will configure static and default routes. A **static route** is a route that is entered manually by the network administrator to create a reliable and safe route. A **directly attached static route** relies on its **exit interface** in order for packets to be sent to its destination, while a **recursive static route** uses the **IP address of the next hop router**.

A **default route**, also known as **the gateway of last resort**, is the network route used by a router when no other known route exists for a destination network. A **static route** is used to route traffic to a **specific network**, while a **default route** is used when destination network is **unknown**.

2.1 Instructions: Static and Default Routes

Step 1: Physical Connections

Create a topology as shown in the following figure. (Given .pkt file)



Step 2: IP Addressing Table

Configure IP addresses of PCs (already configured).

Step 3: Host Name

Set hostnames for routers (R1, R2, and R3) as shown in the topology. (**R1(config)# hostname R1**)

Step 4: Adding IP Addresses

Add IP addresses to both an Ethernet (i.e., G0/0) and serial interface (i.e., S0/0/1). For serial interface with the **DCE cable**, you will need to also add the clocking with the **clock rate** command. **Get the IP addresses from the diagram.**

Step 5: Configure Serial interfaces

Configure **DTE serial interface**: Set IP address, Subnet mask and make interface active (no shutdown).

Configure **DCE serial interface**: Set IP address, Subnet mask and make interface active (no shutdown).

Configure **clock rate**: Router(config-if)#**clock rate 250000**

Step 6: Configure Static & Default Routes

Configure a **recursive static route**:

Syntax: ip route **Dest_Net_Address** **Net_Mask** **Next_Hop_IP_Address**

Example: ip route 172.31.0.0 255.255.255.0 172.31.1.193

Configure a **directly attached static route**:

Syntax: ip route **Dest_Net_Address** **Net_Mask** **Exit_Interface**

Example: ip route 172.31.1.0 255.255.255.128 Serial0/0/0

Configure a **default route**:

Syntax: ip route **0.0.0.0** **0.0.0.0** **Next_Hop_IP_Address**

Example: ip route 0.0.0.0 0.0.0.0 0.0.0.0 Serial0/0/1

Commands for STATIC route in R1 & R3 and DEFAULT route in R3:

R1(config)#ip route 201.169.100.0 255.255.255.0 172.16.100.130

R1(config)#ip route 172.16.101.192 255.255.255.252 172.16.100.130

R1(config)#ip route 10.10.8.0 255.255.255.0 172.16.100.130

R2(config)#ip route 10.10.8.0 255.255.255.0 172.16.101.194

R2(config)#ip route 192.168.100.0 255.255.255.0 172.16.100.129

R2(config)#ip route 192.168.200.0 255.255.255.0 172.16.100.129

R3(config)#ip route 0.0.0.0 0.0.0.0 172.16.101.193

Step 7: Verify Routing table and Test the network (ping host in one network to other networks)

Use the **show ip route** command to verify that each router has all of the networks in the topology entered in the routing table.

When you are finished with the routing configuration, return to **privileged EXEC mode** and save the current configuration to NVRAM.

R1(config-router)#end

%SYS-5-CONFIG_I: Configured from console by console

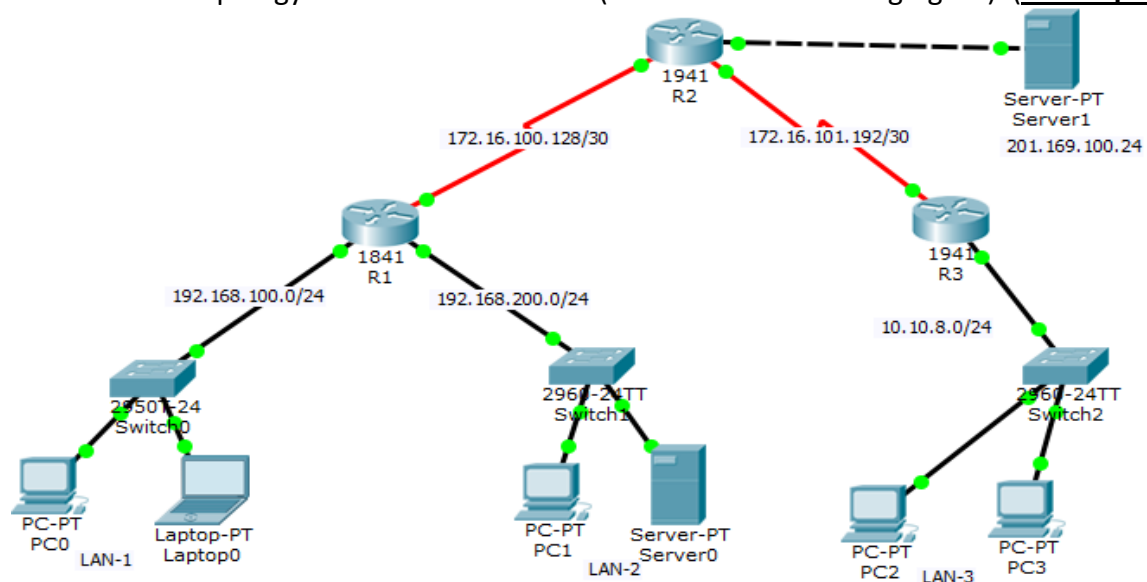
R1#copy run start

3. Background: RIP → A Dynamic Routing Protocol

In this activity, you will configure a simple dynamic routing protocol named **Routing Information Protocol (RIP)**. RIP is a relatively old but still commonly used **interior gateway protocol** created for use in small, homogeneous networks. It is a classical **distance-vector routing protocol**.

3.1 Instructions: Configuring Routing Information Protocol (RIP)

Step 1: Use the same topology as used in **section 2.1** (shown in the following figure). (**Given .pkt file**)



Step 2: IP Addressing Table

Configure IP addresses according to the given diagram (already configured).

Step 3: Adding IP Addresses

Add IP addresses to both an Ethernet (i.e., Fa0/0) and serial interface (i.e., S0/0/1). For serial interface with the **DCE cable** you will need to also add the clocking with the **clock rate** command. **Get the IP addresses from the given diagram.**

Step 4: Adding Dynamic Routing: RIP

For this router to participate in a dynamic routing using a **dynamic routing protocol** like **RIP**, you'll need to enable a routing protocol and advertise the **directly connected networks** that want advertised. To enable a dynamic routing protocol, enter **global configuration mode** and use the **router** command. Enter **router ?** at the global configuration prompt to see a list of available routing protocols on your router. To enable RIP, enter the command **router rip** in global configuration mode.

```
R1(config)#router rip
```

```
R1(config-router)#
```

Once you are in routing configuration mode, enter the network address for each **directly connected network**, using the **network** command (For R1, there are 3 directly connected networks).

```
R1(config-router)#network 192.168.100.0
```

```
R1(config-router)#network 192.168.200.0
```

```
R1(config-router)#network 172.16.100.128
```

```
R1(config-router)#
```

The **network** command:

- Enables RIP on all interfaces that belong to this network. These interfaces will now both send and receive RIP updates.
- Advertises this network in RIP routing updates sent to other routers every **30 seconds**.

When you are finished with the RIP configuration, return to **privileged EXEC mode** and save the current configuration to NVRAM.

```
R1(config-router)#end
```

```
%SYS-5-CONFIG_I: Configured from console by console
```

```
R1#copy run start
```

Step 5: Verify RIP

Use the **show ip route** command to verify that each router has all of the networks in the topology entered in the routing table.

FYI: We need to **advertise the network, not any particular host**. An example of that would be **enabling RIP on ISP**. We want the other router (UIU) to know that any packet destined for the network 192.168.20.0 can be sent to **ISP** which has a directly connected entry in its routing table showing what interface to send the packet to; in this case its **fa0/0**. Check your routing table for entries that are preceded by a capital letter **"R"** to ensure that you are receiving routing updates using RIP. Use **show ip route** to see the routing table. Ensure that both routers configured so that you can receive his updates. **No updates, no ping.**

Demonstrate your work to the instructors before leaving.

Signature of the Instructor

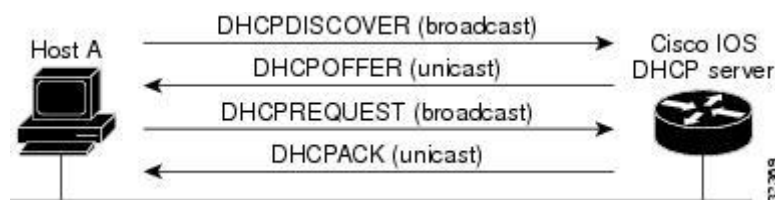
Lab 5: Dynamic Host Configuration Protocol (DHCP), Extending the network with Wireless LAN (Wi-Fi)

1. Objectives

- To learn about **Dynamic Host Configuration Protocol (DHCP)**: why and how used?
- To build an **internetwork** and configure **DHCP** using **Packet Tracer**
- **Configure Wireless router** to create a wireless LAN (Wi-Fi) to extend a wired LAN

2. Background

Dynamic Host Control Protocol (DHCP) enables you to automatically assign reusable IP addresses to DHCP clients. The **DHCP Server** feature is a full DHCP server implementation that assigns and manages IP addresses from specified **address pools** within the router to DHCP clients. **Figure 1** shows the basic steps that occur when a DHCP client requests an IP address from a DHCP server. The client, Host A, sends a **DHCPDISCOVER** broadcast message to locate a **DHCP Server**. A DHCP server offers configuration parameters (such as an IP address, a MAC address, a domain name, and a lease for the IP address) to the client in a **DHCPOFFER** unicast message.



Benefits

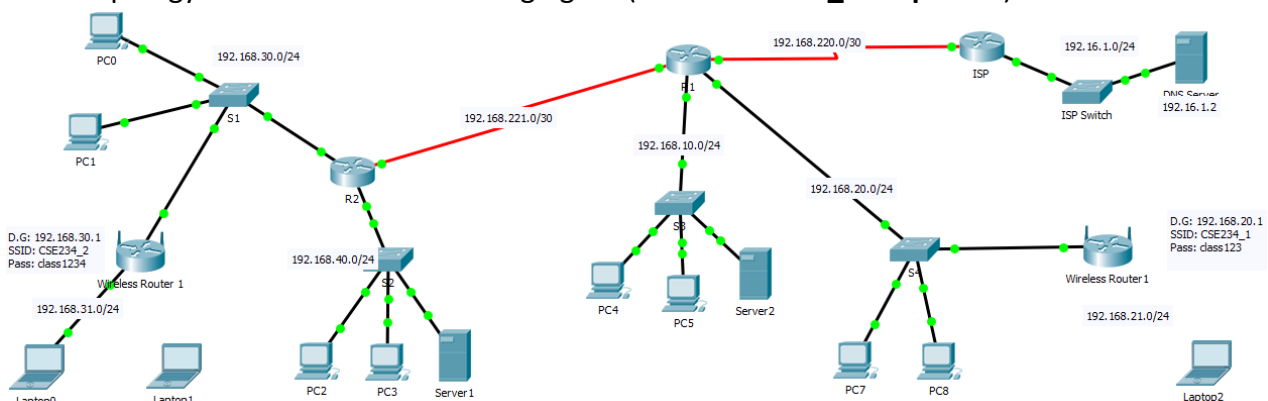
- ✓ **Reduced Internet access costs:** Using automatic IP address assignment at each remote site substantially reduces Internet access costs. Static IP addresses are considerably more expensive to purchase than are automatically allocated IP addresses.
- ✓ **Reduced client configuration tasks and costs:** Because DHCP is easy to configure, it minimizes operational overhead and costs associated with device configuration tasks and eases deployment by nontechnical users.
- ✓ **Centralized management:** Because the DHCP server maintains configurations for several subnets, an administrator only needs to update a single, central server when configuration parameters change.

3. Instructions

This lab provides an opportunity to revise your understanding of DHCP, and the commands for configuring DHCP functions on a router. One router is the **DHCP server**. The other router forwards DHCP requests to the server.

Task 1: Physical Connections

Create a topology as shown in the following figure (Given as **DHCP_WiFi.pkt** file):



Task 2: Configure PC2, PC3, PC4 and PC5 to receive an IP address through DHCP

- ✓ Click on the **PC2**.
- ✓ Click on **Desktop → IP Configuration**
- ✓ Select **DHCP** option to set PC2 to get dynamic IP from the **DHCP Server**.
- ✓ Do the same for **PC3, PC4, and PC5**.

Task 3: Configure a DHCP Server

Configure the **R2 router** as a DHCP server for the **192.168.40.0/24** subnet.

Step 1: Exclude statically assigned addresses.

The DHCP server assumes that all IP addresses in a DHCP address pool subnet are available for assigning to DHCP clients. You must specify the IP addresses that the DHCP server should not assign to clients. These IP addresses are usually static addresses reserved for the router interface, switch management IP address, servers, and local network printer. The **ip dhcp excluded-address** command prevents the router from assigning IP addresses within the configured range. The following commands exclude the **first 10 IP addresses** from each pool for the LANs attached to R1. These addresses will not be assigned to any DHCP clients.

```
R2(config)#ip dhcp excluded-address 192.168.40.1 192.168.40.10
```

Step 2: Configure the pool.

Create the DHCP pool using the **ip dhcp pool** command and name it **R1Fa0**.

```
R2(config)#ip dhcp pool R1Fa0
```

Specify the subnet to use when assigning IP addresses. DHCP pools automatically associate with an interface based on the network statement. The **router R2** now acts as a **DHCP server**, handing out addresses in the **192.168.40.0/24** subnet starting with 192.168.40.11.

```
R2(dhcp-config)#network 192.168.40.0 255.255.255.0
```

Configure the **default router** and **domain name server** for the network. Clients receive these settings via DHCP, along with an IP address.

```
R2(dhcp-config)#dns-server 172.16.17.5
```

```
R2(dhcp-config)#default-router 192.168.40.1
```

Note: There is not a DNS server at 172.16.17.5. You are configuring the command for practice only.

Task 4: Test the DHCP configuration

Configure the **Server1** with the **static** address **192.168.40.2**. It is assumed that the server requires a **fixed IP address** in order to allow access from any PC in the LAN. Configure the server address manually. Ping from the PC2 and PC3 to router ports and server host.

Task 5: Implement DHCP in router R1 for PC4 and PC5 using the network address 192.168.10.0/24

Do yourself.

Demonstrate your work to the instructors before leaving.

Signature of the Instructor

Task 6: Configure a Wireless router to create a Wireless LAN (Wi-Fi)**Part 1: Connect to the Wireless router****Step 1: Establish and verify connectivity to the Wireless router 1.**

- Connect the appropriate cable from **Laptop0** to the Ethernet 1 port on **Wireless router1**.
- Wait for the link light to turn green. Then open the command prompt for **Laptop0**. Use the **ipconfig** command to verify **Host-received** IP addressing information.
- Enter the command **ping 192.168.31.1** to verify **Laptop0** can access the default gateway. (This address of default gateway will be given in the setup manual when you buy a wireless router)

Step 2: Access the wireless router graphical user interface (GUI) using a web browser.

- To configure the **Wireless router 1** using the GUI, you will need to access it using the **Laptop0** web browser. Open the web browser and enter the default gateway address (192.168.31.1) in the URL field.
- Enter **admin** as the default username and password to access the router.

Part 2: Enable Wireless Connectivity**Step 1: Configure the Linksys router for Internet connectivity.**

There is no Internet connectivity in this scenario, but you will still configure the settings for the Internet-facing interface. For **Internet Connection Type**, choose **Static IP** from the drop down list. Then enter the following static IP information:

- ✓ Internet IP Address – **192.168.30.10**
- ✓ Subnet Mask – **255.255.255.0**
- ✓ Default Gateway – **192.168.30.1**
- ✓ DNS 1 – **192.168.1.2**

Step 2: Configure the inside network parameters.

Scroll down to the **Network Setup** section and configure the following information:

- ✓ IP Address – **192.168.31.1**
- ✓ Subnet Mask – **255.255.255.0**
- ✓ Starting IP Address – Enter **100** for the last octet.
- ✓ Maximum number of Users – **50**

Note: The IP address range of the DHCP pool will only reflect the changes once you click '**Save Settings**'

Step 3: Save the settings and reconnect to the router.

- Scroll to the bottom of the page and click **Save Settings**. If you move from one tab to another without saving, your configurations will be lost.
- You lose your connection when you click **Save Settings**. This occurred because you changed the IP address of the router.
- Return to the command prompt of **Laptop0**. Enter the command **ipconfig /renew** to renew the IP address.
- Use the **Laptop0** web browser to reconnect to the **router**. You will need to use the new default gateway address. Verify the **Internet Connection** settings in the **Status** tab. The settings should match the values you configured in Part 2, Step 1. If not, repeat Part 2, Step 1 and Step 2.

Step 4: Configure wireless connectivity for wireless devices.

- Click the **Wireless** tab and investigate the options in the dropdown list for **Network Mode**.
- Set the network mode for **Wireless-N Only**.
- Change the SSID to **CSE324_1**.
- When a **wireless client** surveys the area searching for wireless networks, it detects any SSID broadcasts. **SSID broadcasts are enabled by default**.
- For best performance in a network using Wireless-N, set the radio band to **Wide-40MHz**.
- Click **Save Settings** and then click **Continue**.

Step 5: Configure wireless security so that clients must authenticate to connect to the wireless network.

- a. Click the **Wireless Security** option under the **Wireless** tab.
- b. Set the **Security Mode** to **WPA2 Personal**.
- c. Leave the encryption mode to AES and set the passphrase to **class1231**.
- d. Click **Save Settings** and then click **Continue**.

Step 6: Change the default password to access the wireless router for configuration.

- a. You should always change the default password. Click the **Administration** tab and change the **Router Access** password to **cse1234**.
- b. Click **Save Settings**. Enter the username **admin** and the new password.

Part 3: Configure and Verify Wireless Client Access**Step 1:** Configure Laptop1 to access the wireless network.

- a. Click **Laptop1** and click **Desktop > PC Wireless**.
- b. If wireless interface is not found, add a wireless interface (**Laptop1** → **Physical** → **free slot** → **WPC300N** → **add**).
- c. Click **Desktop > PC Wireless**.
- d. Click the **Connect** tab and click **Refresh**, if necessary. You should see **CSE324_1** listed under **Wireless Network Name**.
- e. Click **CSE324_1** and click **Connect**.
- f. The **Pre-shared Key** is the password you configured in Part 2, Step 5c. Enter the password and click **Connect**.
- h. Close the GUI and click **Command Prompt**. Enter the command **ipconfig** to verify **Laptop1** received IP addressing.

Step 2: Verify connectivity between Laptop1 and PC0.

- a. Ping the **Wireless Router 1** from the **Laptop1**.
- b. Ping **PC0** from the **Laptop1**.

Task 7 – Exercise: Configure Wireless router 2 to create a **Wireless LAN (Wi-Fi)** with SSID **CSE324_2**.

Demonstrate your work to the instructors before leaving.

Signature of the Instructor

Lab 6: Packet-level Firewalls – Access Control Lists (ACLs)

1. Objectives

- Use of **Access Control Lists (ACLs)** defined in Routers to control access in a network.
- Difference between **Standard** and **Extended ACLs**.
- Configuring and applying **Standard** and **Extended ACLs** in Cisco routers.

2a. Access Control Lists (ACLs)

The **Access Control List (ACL)** is a collection of security rules or policies that allows or denies packets after looking at the packet headers and other attributes. Each **permit** or **deny** statement in the ACL is referred to as an **access control entry (ACE)**. These ACEs can classify packets by **inspecting Layer 2 through Layer 4 headers** for a number of parameters, including the following:

- ✓ **Layer 2 header** information such as **EtherTypes**
- ✓ **Layer 3 header** information such as **ICMP, TCP, or UDP**
- ✓ **Layer 3 header** information such as **source** and **destination IP addresses**
- ✓ **Layer 4 header** information such as **source** and **destination TCP or UDP ports**

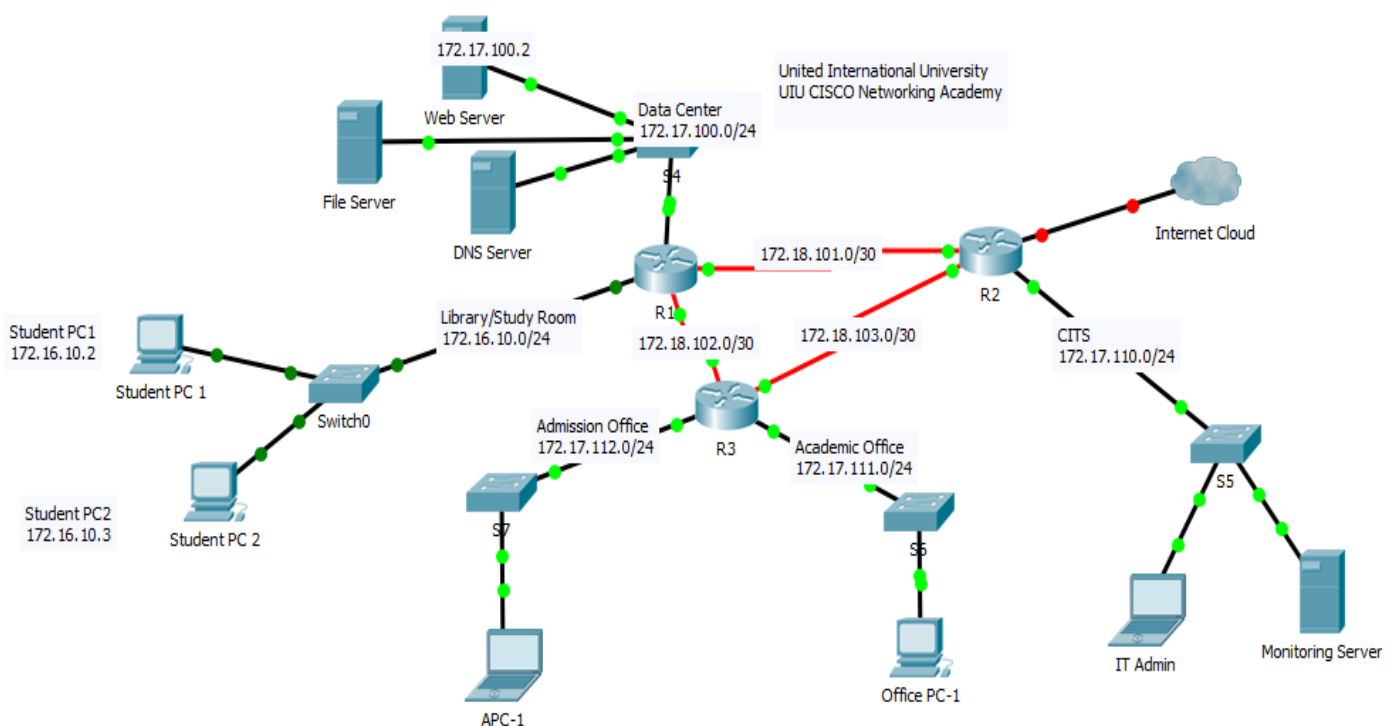
After an ACL has been properly configured, you can **apply it to an interface** to filter traffic. The security appliance can filter packets in both the **inbound** and **outbound direction** on an interface. When an **inbound ACL** is applied to an interface, the security appliance analyzes packets against the ACEs after **receiving them**. If a packet is permitted by the ACL, the firewall continues to process the packet and eventually passes the packet to the defined interface.

2b. Standard ACLs

Standard access control lists (ACLs) are router configuration scripts that control whether a router permits or denies packets based on the **source address only**. Tasks are: **defining filtering criteria**, **configuring standard ACLs**, **applying ACLs to router interfaces**, and **verifying and testing the ACL implementation**.

Example:

Consider the following topology (Given .pkt file):



Create an ACL using the number 1 on **R3 (Why not on R1?)** with a statement that **denies** access to the **Admission Office (172.17.112.0/24)** network from the **Library/Study Room (172.16.10.0/24)** network.

```
R3(config)# access-list 1 deny 172.16.10.0 0.0.0.255
```

- b. By default, an access list denies all traffic that does not match a rule. To permit all other traffic, configure the following statement:

```
R3(config)# access-list 1 permit any
```

- c. For the ACL to actually filter traffic, it must be applied to some router operation. Apply the ACL by placing it for outbound traffic on the Gigabit Ethernet 0/0 interface.

```
R3(config)# interface GigabitEthernet0/0
```

```
R3(config-if)# ip access-group 1 out
```

2c. Extended ACLs

Extended access control lists (ACLs) are extremely powerful. They offer a much greater degree of control than standard ACLs as to the types of traffic that can be filtered, as well as where the traffic originated and where it is going. Extended ACLs can filter traffic in many different ways.

Extended ACLs can filter on **source IP addresses, source ports, destination IP addresses, destination ports**, as well as **various protocols and services**. Tasks are: **defining filtering criteria, configuring extended ACLs, applying ACLs to router interfaces, and verifying and testing the ACL implementation.**

- a. Create two access list statements to **permit tcp** for accessing **FTP server at 172.17.100.3** and **permit ICMP (ping, etc.)** traffic from **Library/Study Room (172.16.10.0/24)** network to **Server Farm at 172.17.100.0/24**. Note that the **access list number remains the same** and a specific type of ICMP traffic does not need to be specified.

```
R1(config)# access-list 100 permit icmp 172.16.10.0 0.0.0.255 host 172.17.100.3
```

```
R1(config)# access-list 100 permit tcp 172.16.10.0 0.0.0.255 172.17.100.0 0.0.0.255 eq ftp
```

- b. Enter interface configuration mode and apply the ACL.

```
R1(config)# interface gigabitEthernet 0/0
```

```
R1(config-if)# ip access-group 100 in
```

Demonstrate your work to the instructors before leaving.

Signature of the Instructor

Lab 7: Network Address Translation (NAT) – Connecting to an ISP (Internet)

1. Objectives

- To learn about **Network Address Translation (NAT)**: why and how used?
- Types of NAT: **Static** and **Dynamic NAT, Port Address Translation (PAT)**
- To build an **internetwork** using NAT using **Packet Tracer**

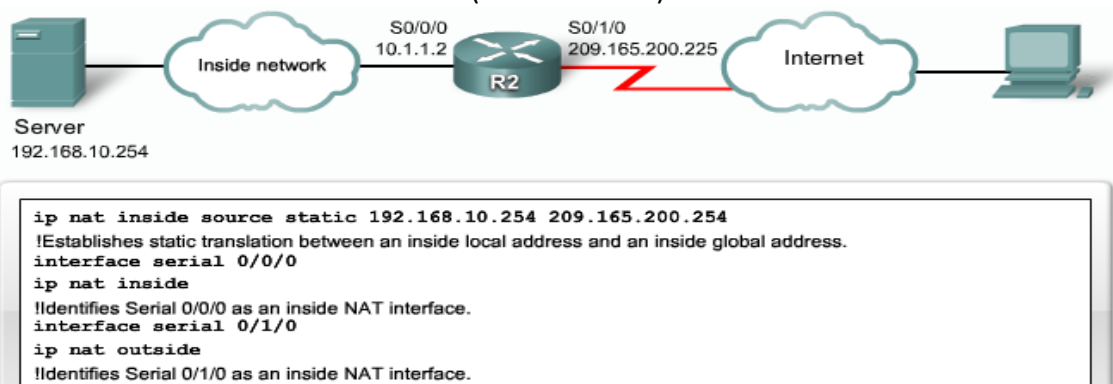
2. Background: Network Address Translation (NAT)

NAT (Network Address Translation) is a technique for preserving scarce Internet IP addresses. It converts private IP addresses (not routable to Internet) to public IP addresses so that Internet can be accessed from private network.

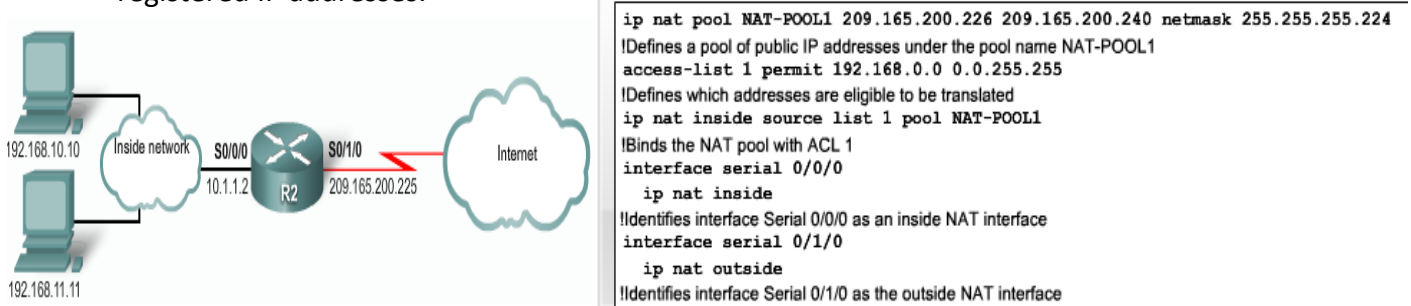
Types of NAT

Developed by Cisco, Network Address Translation is used by a device (firewall, router or computer) that sits between an internal network and the rest of the world. NAT has many forms and can work in several ways:

- ✓ **Static NAT** - Mapping an unregistered (private) IP address to a registered (public) IP address on a one-to-one basis. Particularly useful when a local host with private IP address needs to be accessible from outside the network (from Internet).



- ✓ **Dynamic NAT** - Maps an unregistered IP address to a registered IP address from a group of registered IP addresses.



- ✓ **Overloading** - A form of dynamic NAT that maps multiple unregistered IP addresses to a single registered IP address by using different ports. This is known also as **PAT (Port Address Translation)**, single address NAT or port-level multiplexed NAT.

```

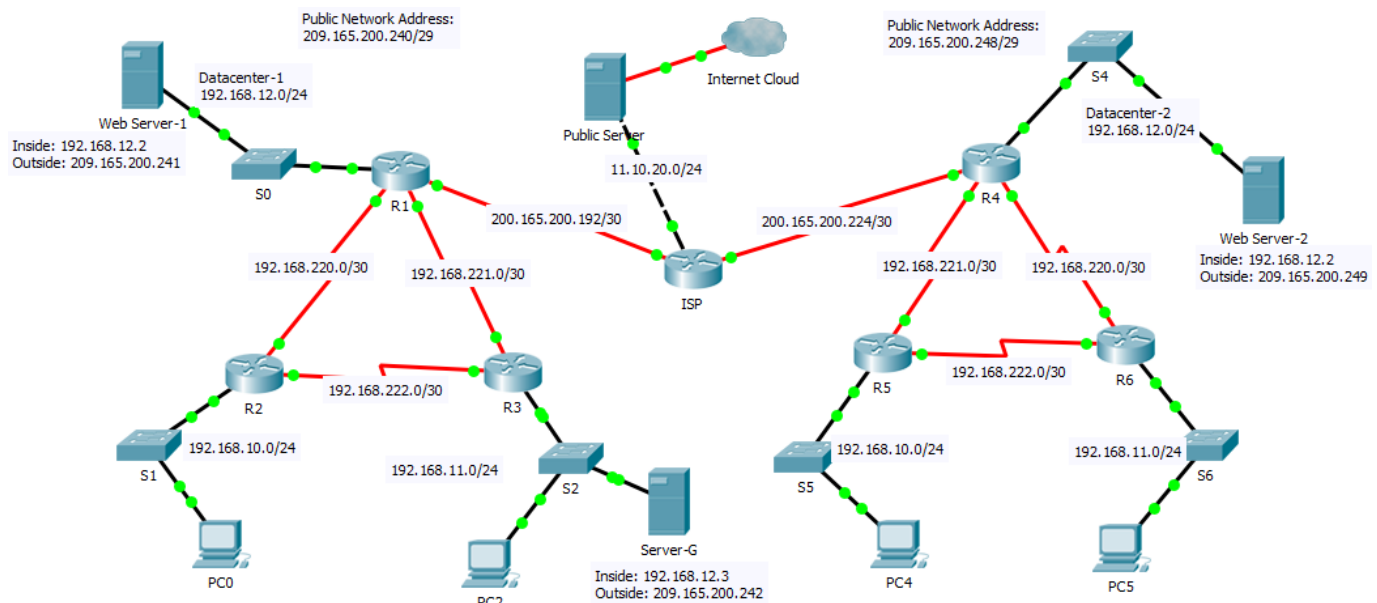
access-list 1 permit 192.168.0.0 0.0.255.255
!Defines which addresses are eligible to be translated
ip nat inside source list 1 interface serial 0/1/0 overload
!Identifies the outside interface Serial 0/1/0 as the inside global address to be overloaded
interface serial 0/0/0
ip nat inside
!Identifies Serial 0/0/0 as an inside NAT interface.
interface serial 0/1/0
ip nat outside
!Identifies Serial 0/1/0 as an inside NAT interface.
  
```

3a. Instructions:

This lab provides an opportunity to revise your understanding of NAT/PAT, and the commands for configuring NAT/PAT on a router. The router R2 translates private IP to public IP using NAT/PAT.

Task 1: Create the Topology

Create a topology as shown in the following figure (Given .pkt file):

**Task 2: Configure Static and Default Routing**

ISP (Internet Service Provider) uses **static routing** to reach all networks beyond R1 and R4 (Border routers). However, R1 and R4 translate **private addresses** into **public addresses** before sending traffic to ISP (Internet). Therefore, ISP must be configured with the public addresses (**209.165.200.240/29** for R1 and **209.165.200.248/29** for R4) that are part of the NAT configurations on R1 and R4. We will show the **configurations on R1**, configurations on R4 will be left as an exercise.

Enter the following **static route** on ISP:

```
ISP(config)#ip route 209.165.200.240 255.255.255.248 serial 0/0/0
```

This static route includes all addresses assigned to R1 for public use. Configure a **default route** on R1 and **propagate** the route in RIP.

```
R1(config)#ip route 0.0.0.0 0.0.0.0 209.165.200.193
```

```
R1(config)#router rip
```

```
R1(config-router)#default-information originate
```

Allow a few seconds for R2 & R3 to learn the **default route from R1** and then check the R2 & R3 routing table using **show ip route** command. Alternatively, you can clear the routing table with the **clear ip route *** command. A default route pointing to R1 should appear in the R2 & R3 routing table. From R2, ping the **serial 0/1/1 interface** (connected to ISP) on R1 (209.165.200.194). The pings should be successful. Troubleshoot if the pings fail.

Task 3: Configure Static NAT**Step 1: Statically map a public IP address to a private IP address.**

The **Web Server 1** attached to R1 is accessible by outside hosts beyond ISP. Statically assign the public IP address **209.165.200.241** as the address for NAT to use to map packets to the private IP address of the **Web Server 1** at **192.168.12.2**. Similarly, assign the public IP address **209.165.200.242** to the private IP address of the **Server-G** at **192.168.11.3**.

```
R1(config)#ip nat inside source static 192.168.12.2 209.165.200.241
```

```
R1(config)#ip nat inside source static 192.168.11.3 209.165.200.242
```

Step 2: Specify inside and outside NAT interfaces.

Before NAT can work, you must specify which interfaces are inside and which interfaces are outside.

```
R1(config)#interface serial 0/1/1
R1(config-if)#ip nat outside
R1(config-if)#interface G0/0
R1(config-if)#ip nat inside
```

Step 3: Verify the static NAT configuration.

From ISP (or Public Server), ping the public IP address **209.165.200.241** and **209.165.200.242**.

Task 4: Configure Dynamic NAT with a Pool of Addresses

While static NAT provides a permanent mapping between an internal address and a specific public address, dynamic NAT maps private IP addresses to public addresses. These public IP addresses come from a NAT pool.

Step 1: Define a pool of global addresses.

Create a pool of addresses to which matched source addresses are translated. The following command creates a pool named **MY-NAT-POOL** that translates matched addresses to an available IP address in the **209.165.200.243 - 209.165.200.246** range.

```
R1(config)#ip nat pool MY-NAT-POOL 209.165.200.243 209.165.200.246 netmask 255.255.255.248
```

Step 2: Create a standard access control list to identify which inside addresses are translated.

```
R1(config)#access-list 1 permit 192.168.10.0 0.0.0.255
R1(config)#access-list 1 permit 192.168.11.0 0.0.0.255
```

Step 3: Establish dynamic source translation by binding the pool with the access control list.

A router can have more than one NAT pool and more than one ACL. The following command tells the router which address pool to use to translate hosts that are allowed by the ACL.

```
R1(config)#ip nat inside source list 1 NAT pool MY-NAT-POOL
```

Step 4: Specify inside and outside NAT interfaces.

You have **already specified** the inside and outside interfaces for your static NAT configuration. Now add other serial interfaces linked to R1 as inside interfaces.

```
R1(config)#interface serial 0/0/0
R1(config-if)#ip nat inside
R1(config)#interface serial 0/0/1
R1(config-if)#ip nat inside
```

Step 5: Verify the configuration.

Ping ISP from PC0 and PC2. Then use the **show ip nat translations** command on R1 to verify NAT.

```
R1#show ip nat translations
```

Pro	Inside global	Inside local	Outside local	Outside global
---	209.165.200.241	192.168.12.2	---	--
---	209.165.200.242	192.168.11.3	---	---
---	209.165.200.243	192.168.10.2	200.165.200.193	200.165.200.193
---	209.165.200.244	192.168.11.2	200.165.200.193	200.165.200.193

Task 5: Configure NAT Overload

In the previous example, what would happen if you needed more than the **four public IP addresses** that the pool allows? By tracking **port numbers**, **NAT overloading** allows **multiple inside users to reuse a public IP address**. In this task, you will remove the pool and mapping statement configured in the previous task. Then you will configure NAT overload on R1 so that all internal IP addresses are translated to the **R1 S0/1/1 IP address** when connecting to any outside device.

Step 1: Remove the NAT pool and mapping statement.

Use the following commands to remove the NAT pool and the map to the NAT ACL.

```
R1(config)#no ip nat pool MY-NAT-POOL 209.165.200.243 209.165.200.246 netmask 255.255.255.248
```

```
R1(config)#no ip nat inside source list 1 pool MY-NAT-POOL
```

If you receive the following message, clear your NAT translations.

```
%Pool MY-NAT-POOL in use, cannot destroy
```

```
R1#clear ip nat translation *
```

Step 2: Configure PAT on R1 using the serial 0/1/1 (outside) interface public IP address.

The configuration is similar to dynamic NAT, except that instead of a pool of addresses, the **interface** keyword is used to identify the outside IP address. Therefore, **no NAT pool is defined**. The **overload** keyword enables the addition of the port number to the translation. Because you already configured an ACL to identify which inside IP addresses to translate as well as which interfaces are inside and outside, you only need to configure the following:

```
R1(config)#ip nat inside source list 1 interface S0/1/1 overload
```

Step 3: Verify the configuration.

Ping ISP from PC0 and PC2. Then use the **show ip nat translations** command on R1 to verify NAT.

```
R2#show ip nat translations
```

Pro	Inside global	Inside local	Outside local	Outside global
icmp	209.165.200.194:3	192.168.10.11:3	209.165.200.193:3	209.165.200.193:3
icmp	209.165.200.194:1024	192.168.11.11:3	209.165.200.193:3	209.165.200.193:1024
---	209.165.200.241	192.168.12.2	---	---
---	209.165.200.242	192.168.11.3	---	---

Note: In the previous task, you could have added the keyword **overload** to the **ip nat inside source list 1 pool MY-NAT-POOL** command to allow for more than four concurrent users.

Task 6 – Exercise: Configure STATIC NAT (to access Web Server-2) and DYNAMIC NAT (to access INTERNET from the LANs connected to R4) in the router R4.

Demonstrate your work to the instructors before leaving.

Signature of the Instructor

Lab 8: Introduction to Socket Programming

1. Objectives

- Introduction to **Socket Programming** concepts.
- Introduction to **Python API** for Socket Programming.
- To create a **Client-Server application** using **Socket Programming** in **Python**.

2. Background

2a. Socket Basics

A *network socket* is an endpoint of an inter-process communication flow across a computer network. Sockets may communicate within a process, between processes on the same machine, or between processes on different continents. Today, most communication between computers is based on the internet protocol; therefore most network sockets are *internet sockets*. To create a connection between machines, Python programs import the **socket** module, create a socket object, and call the object's methods to establish connections and send and receive data. Sockets are the endpoints of a bidirectional communications channel.

In this lab, you will learn the basics of socket programming for TCP connections in Python: how to create a socket, bind it to a specific address and port, as well as send and receive a HTTP packet. You will also learn some basics of HTTP header format.

2b. Hints

Here are a few hints that may help you as you write the program.

- ✓ You have to choose a **server port** to connect to. Ports from 1-1023 are mostly used for certain services and require administrative privileges. Use port numbers greater than at least 1023.
- ✓ **Close** your sockets cleanly before exit. If you abort the program, the port may not be freed.
- ✓ You can run all of the processes on the same machine. For your machine, just use localhost. You can use **ifconfig (unix)** or **ipconfig (windows)** to know IP address for testing multiple machines.
- ✓ Be wary of overzealous firewalls stopping your connections - try **temporarily disabling firewalls** if you find your connections timeout or are denied.

2c. Create a Socket

This first thing to do is create a socket. The **socket.socket** function does this. Run **socketExample1.py**. The code will create a socket with **Address Family** : **AF_INET** (this is IP version 4 or IPv4), **Type** : **SOCK_STREAM** (this means connection oriented TCP protocol).

2d. Connect to a Server

We connect to a remote server on a certain port number. So we need 2 things , IP address and port number to connect to. So you need to know the IP address of the remote server you are connecting to. Here we used the ip address of **google.com** as a sample. Run **socketExample2.py**. It creates a socket and then connects. Try connecting to a port different from port 80 and you should not be able to connect which indicates that the port is not open for connection. This logic can be used to build a port scanner.

2e. Sending data to a Server

Function **sendall** will simply send data. Let us send some data to google.com. Run **socketExample3.py**. In the above example , we first connect to an ip address and then send the string message "**GET / HTTP/1.1\r\n\r\n**" to it. The message is actually an "**http command**" to fetch the **mainpage of a website**. Now that we have send some data, it's time to receive a reply from the server. So let us do it.

2f. Sending data to a Server and receiving data from the Server

Function **recv** is used to receive data on a socket. In the following example we shall send the same message as the last example and receive a reply from the server. Run **socketExample4.py**. Google.com replied with the content of the page we requested. Quite simple! Finally, we close the socket.

3. Sample Server and Client Code

3a. A Simple Server

To write Internet servers, we use the **socket** function available in socket module to create a socket object. A socket object is then used to call other functions to setup a socket server. Now call **bind(hostname, port)** function to specify a *port* for your service on the given host. Next, call the *accept* method of the returned object. This method waits until a client connects to the port you specified, and then returns a *connection* object that represents the connection to that client.

```
# server.py
# TCP Server Code

host="127.0.0.1"    # Set the server address to variable host
port=4444          # Sets the variable port to 4444

from socket import * # Imports socket module
s=socket(AF_INET, SOCK_STREAM)
s.bind((host,port))  # Binds the socket. Note that the input to
                    # the bind function is a tuple
s.listen(1)          # Sets socket to listening with a queue
                    # of 1 connection
print "Listening for connections.. "
q,addr=s.accept()    # Accepts incoming request from client and returns
                    # socket and address to variables q and addr
data=raw_input("Enter data to be send: ")
                    # Data to be send is stored in variable data from user

q.send(data)         # Sends data to client
s.close()

# End of code
```

3b. A Simple Client

Now we will write a very simple client program which will open a connection to a given port 12345 and given host. This is very simple to create a socket client using Python's *socket* module function. The **socket.connect(hostname, port)** opens a TCP connection to *hostname* on the *port*. Once you have a socket open, you can read from it like any IO object. When done, remember to close it, as you would close a file. The following code is a very simple client that connects to a given host and port, reads any available data from the socket, and then exits:

```
#Client
# TCP Client Code

host="127.0.0.1"    # Set the server address to variable host
port=4444          # Sets the variable port to 4444

from socket import * # Imports socket module
s=socket(AF_INET, SOCK_STREAM) # Creates a socket
s.connect((host,port))    # Connect to server address
msg=s.recv(1024) # Receives data upto 1024 bytes, stores in a var msg
print ("Message from server : " + msg.strip().decode('ascii'))
s.close()             # Closes the socket

# End of code
```

Now run this **server.py** in background and then run above **client.py** to see the result.

Output:

Step 1: Run **server.py**. It would start a server in background.

Step 2: Run **client.py**. Once server is started run client.

Step 3: Output of **server.py** generates as follows:

```
C:\PythonDir\>python server.py
Listening for connections..
Enter data to be send: cse323
```

Step 4: Output of **client.py** generates as follows:

```
C:\ PythonDir\>python clients.py
Message from server : cse323
```

4. Exercise: A web Server

You will develop a **web server** that handles **one HTTP request** at a time. Your web server should accept and parse the HTTP request, get the requested file from the server's file system, create an **HTTP response message** consisting of the requested file preceded by header lines, and then send the response directly to the client. If the requested file is not present in the server, the server should send an HTTP **"404 Not Found"** message back to the client.

4a. Skeleton Code

Below find the skeleton code for the Web server. You are to **complete the skeleton code**. The places where you need to fill in code are marked with **#Fill in start** and **#Fill in end**. Each place may require one or more lines of code.

4b. Running the Server

Put an **HTML file** (e.g., HelloWorld.html) in the **same directory** that the server is in. Run the server program. Determine the IP address of the host that is running the server (e.g., 128.238.251.26). From another host, open a browser and provide the corresponding URL. For example:

http://128.238.251.26:6789/HelloWorld.html

'HelloWorld.html' is the name of the file you placed in the server directory. Note also the use of the port number after the colon. You need to **replace this port number** with whatever port you have used in the server code. In the above example, we have used the port number **6789**. The browser should then display the contents of **HelloWorld.html**. If you omit ":6789", the browser will assume port 80 and you will get the web page from the server only if your server is listening at port 80. Then try to get a file that is not present at the server. You should get a "404 Not Found" message.

4c. Submit the files

You will hand in the **complete server code** along with the **screen shots of your client browser**, verifying that you actually receive the contents of the HTML file from the server.

Skeleton Python Code for the Web Server

```
#import socket module
from socket import *
serverSocket = socket(AF_INET, SOCK_STREAM)
#Prepare a sever socket
#Fill in start
#Fill in end
while True:
    #Establish the connection
    print 'Ready to serve...'
    connectionSocket, addr = #Fill in start #Fill in end
```



```
try:
    message = #Fill in start #Fill in end
    filename = message.split()[1]
    f = open(filename[1:])
    outputdata = #Fill in start #Fill in end
    #Send one HTTP header line into socket
    #Fill in start
    #Fill in end
    #Send the content of the requested file to the client
    for i in range(0, len(outputdata)):
        connectionSocket.send(outputdata[i])
    connectionSocket.close()
except IOError:
    #Send response message for file not found
    #Fill in start
    #Fill in end
    #Close client socket
    #Fill in start
    #Fill in end
    serverSocket.close()
```

5. Optional Exercises (Extra Credit up to 3)

1. Currently, the web server handles **only one HTTP request at a time**. Implement a **multithreaded server** that is capable of serving **multiple requests simultaneously**. Using threading, first create a **main thread** in which your modified server listens for clients at a fixed port. When it receives a TCP connection request from a client, it will set up the TCP connection through another port and services the client request in a separate thread. There will be a separate TCP connection in a separate thread for each request/response pair.

2. Instead of using a browser, **write your own HTTP client** to test your server. Your client will connect to the server using a TCP connection, send an HTTP request to the server, and display the server response as an output. You can assume that the **HTTP request sent is a GET method**.

The client should take **command line arguments** specifying the **server IP address or host name**, **the port** at which the server is listening, and **the path** at which the requested object is stored at the server. The following is an input command format to run the client.

```
client.py server_host server_port filename
```

Demonstrate your work to the instructors before leaving.

Signature of the Instructor