**COMPUTER LABORATORY MANUAL**

|  |  |
| --- | --- |
|  |  |

**Computer and Communication Networks**

**(CSC - 313 L)**

**Version 1.2**

**DEPARTMENT OF SOFTWARE ENGINEERING (DSE)**

**FOUNDATION UNIVERSITY RAWALPINDI CAMPUS (FURC)**

**[www.fui.edu.pk](http://www.fui.edu.pk)**

**<http://www.fui.edu.pk/FURC/>**

**PREFACE**

This lab manual has been prepared to facilitate the students of software engineering in studying and analysing various functions of a computer network. The students will have plan the IP address scheme, configure and test the several network devices. Different tools are used to monitor network traffic and analyse packets. The lab sessions are designed to improve the abilities of the students by giving hands on experience. After completing the laboratory exercises, the students will be familiar with the practical issues of the different concepts explained in the course, as well as with the real equipment used nowadays in computer networks.

**PREPARED BY**

Lab manual is prepared by Mr. Muhammad Usman Khan, Ms. Tehmina Karamat and Asst. Prof. Umar Mahmud under the supervision of Head of Department Dr. Muhammad Shaheen.

**GENERAL INSTRUCTIONS**

1. Students are required to maintain the lab manual with them till the end of the semester.
2. All readings, answers to questions and illustrations must be solved on the place provided. If more space is required then additional sheets may be attached. You may add screen print to the report by using the ‘Print Screen’ command on your keyboard to get a snapshot of the displayed output.
3. It is the responsibility of the student to have the manual graded before deadlines as given by the instructor.
4. Loss of manual will result in re submission of the complete manual.
5. Students are required to go through the experiment before coming to the lab session.
6. Students must bring the manual in each lab.
7. Keep the manual neat clean and presentable.
8. Plagiarism is strictly forbidden. No credit will be given if a lab session is plagiarised and no re-submission will be entertained.
9. Marks will be deducted for late submission.
10. You need to submit the report even if you have demonstrated the exercises to the lab instructor or shown them the lab report during the lab session.

**VERSION HISTORY**

|  |  |  |
| --- | --- | --- |
| **Date** | **Update By** | **Details** |
| July 2013 | Muhammad Usman Khan | Version 1.0. Initial draft prepared and experiments outlined. |
| Sept 2013 | Tehmina Karamat | Version 1.1. Improvements added in the manual |
| Sept 2013 | Umar Mahmud | Version 1.2. Formatting corrected. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**MARKS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **LAB #** | **Date Conducted** | **Lab Title** | **Max. Marks** | **Marks Obtained** | **Instructor Sign** |
| **1** |  | **INTRODUCTION TO NETWORK DEVICES** | **10** |  |  |
| **2** |  | **TRANSMISSION MEDIA** | **10** |  |  |
| **3** |  | **CLASSIFICATION OF IP** | **10** |  |  |
| **4** |  | **CONFIGURING IP** | **10** |  |  |
| **5** |  | **BASIC NETWORKING COMMANDS IN MSDOS** | **10** |  |  |
| **6** |  | **CONFIGURING DHCP AND DNS** | **10** |  |  |
| **7** |  | **SUBNETTING** | **10** |  |  |
| **8** |  | **SUBNETTING CLASS A AND B** | **10** |  |  |
| **9** |  | **PACKET TRACER AND DESIGNING TOPOLOGY** | **10** |  |  |
| **10** |  | **ROUTER CONFIGURATION** | **10** |  |  |
| **11** |  | **STATIC ROUTING** | **10** |  |  |
| **12** |  | **DYNAMIC ROUTING** | **10** |  |  |
| **13** |  | **ACCESS CONTROL LIST IN ROUTER** | **10** |  |  |
| **14** |  | **SOCKET AND SERVER SOCKET PROGRAMMING** | **10** |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| **Grand Total** | | |  |  |  |

LIST OF EXPERIMENTS

[EXPERIMENT 1 – INTRODUCTION TO NETWORK DEVICES 5](#_Toc462222822)

[EXPERIMENT 2 – TRANSMISSION MEDIA 7](#_Toc462222823)

[EXPERIMENT 3 – CLASSIFICATION OF IP 10](#_Toc462222824)

[EXPERIMENT 4 – CONFIGURING IP 13](#_Toc462222825)

[EXPERIMENT 5 - BASIC NETWORKING COMMANDS IN MSDOS 16](#_Toc462222826)

[EXPERIMENT 6 - CONFIGURING DHCP AND DNS 21](#_Toc462222827)

[EXPERIMENT 7 - SUBNETTING 25](#_Toc462222828)

[EXPERIMENT 8 – SUBNETTING CLASS A AND B 29](#_Toc462222829)

[EXPERIMENT 9 – PACKET TRACER AND DESIGNING TOPOLOGY 33](#_Toc462222830)

[EXPERIMENT 10 – ROUTER CONFIGURATION 39](#_Toc462222831)

[EXPERIMENT 11 – STATIC ROUTING 42](#_Toc462222832)

[EXPERIMENT 12 – DYNAMIC ROUTING 45](#_Toc462222833)

[EXPERIMENT 13 – ACCESS CONTROL LIST IN ROUTER 48](#_Toc462222834)

[EXPERIMENT 14 – SOCKET AND SERVER SOCKET PROGRAMMING 50](#_Toc462222835)

# EXPERIMENT 1 – INTRODUCTION TO NETWORK DEVICES

**Objective**

* Lab structure orientation
* Study Network Devices

**Time Required** : 3 hrs

**Programming Language** : NIL

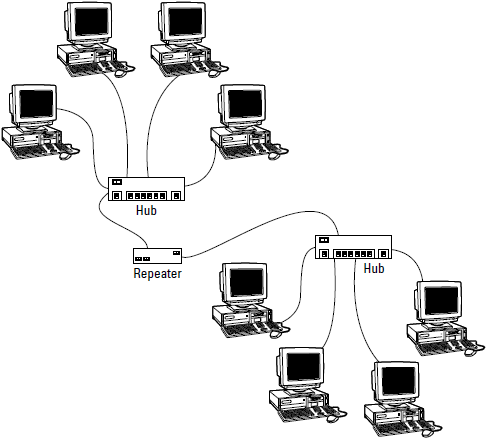
**Software Required** : NIL

**Hardware Required**  :

* Computer with administrative rights

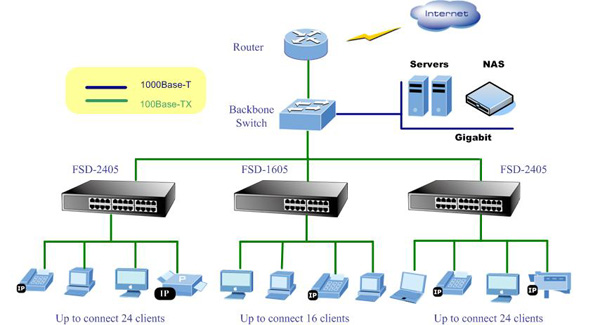
**Structure of the lab**

* IT Lab consists of almost 50 computers connected each other using star topology. Further this Lab is connected with central point (server) through switch.
* IP addresses used in this lab are Class B IP Address i.e. [172.26.x.x] and the subnet mask for this lab is [255.255.0.0].
* Internet service is accessible round the clock which is shared from the central point.

**Network Devices**

**Repeater:** Functioning at Physical Layer. A repeater is an electronic device that receives a signal and retransmits it at a higher level and/or higher power, or onto the other side of an obstruction, so that the signal can cover longer distances. Repeater has two ports, so cannot be use to connect for more than two devices.

**Hub:** An Ethernet hub or concentrator is a device for connecting multiple twisted pair or fiber optic Ethernet devices together and making them act as a single network segment. Hubs work at the physical layer (layer 1) of the OSI model. The device is a form of multiport repeater. Repeater hubs also participate in collision detection, forwarding a jam signal to all ports if it detects a collision.

**Switch:** A network switch or switching hub is a computer networking device that connects network segments. It routes data at the data link layer (layer 2) of the OSI model. Switches that additionally process data at the network layer (layer 3 and above) are often referred to as Layer 3 switches or multilayer switches.

**Router:** A router is an electronic device that interconnects two or more computer networks, and selectively interchanges packets of data between them. Each data packet contains address information that a router can use to determine if the source and destination are on the same network, or if the data packet must be transferred from one network to another. Where multiple routers are used in a large collection of interconnected networks, the routers exchange information about target system addresses, so that each router can build up a table showing the preferred paths between any two systems on the interconnected networks.

**Gate Way:** In a communications network, a network node equipped for interfacing with another network that uses different protocols.

**EXERCISES**

**Exercise 1.1 [2]**

What is the difference between switch and router?

**Exercise 1.2 [2]**

A router is used to connect different \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_?

**Exercise 1.3 [2]**

Write the IP address of your computer.

**Exercise 1.4 [2]**

Which network device is used to connect the computers in LAB?

**Exercise 1.5 [2]**

Analyse the lab network configuration and comment about the topology?

# EXPERIMENT 2 – TRANSMISSION MEDIA

**Objective**

* Study of different types of Network cables and practically implement the cross-wired cable and straight through cable using clamping tool.

**Time Required** : 3 hrs

**Programming Language** : NIL

**Software Required** : NIL

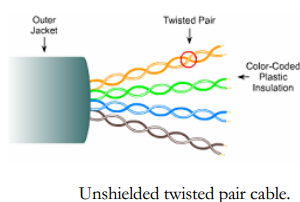
**Hardware Required** :

* UTP Wire
* Crimping Tool
* Connector

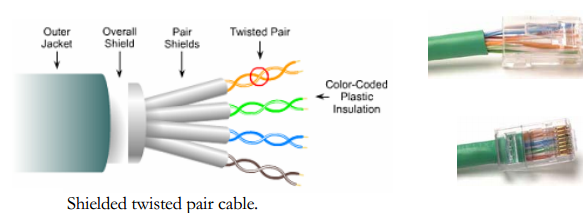
**Network Cable**

There are many types of network cables used in the real-world applications. Some of them are given below:

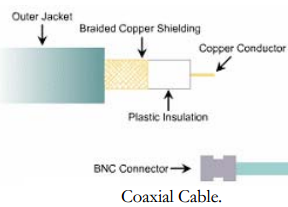
**Unshielded twisted pair:** As the name indicates, the wires are twisted with one another and there is no shield.



**Shielded twisted pair:** Shield with twisted pair.



**Coaxial cable:** Similar to our cable TV cables.



**Implement the cross-wired cable and straight through cable**

* Start by stripping off about 2 inches of the plastic jacket off the end of the cable. Be very careful at this point, as to not nick or cut into the wires, which are inside. Doing so could alter the characteristics of your cable, or even worse render is useless. Check the wires, one more time for nicks or cuts. If there are any, just whack the whole end off, and start over.
* Spread the wires apart, but be sure to hold onto the base of the jacket with your other hand. You do not want the wires to become untwisted down inside the jacket. Category 5 cable must only have 1/2 of an inch of 'untwisted' wire at the end; otherwise it will be 'out of spec'. At this point, you obviously have ALOT more than 1/2 of an inch of un-twisted wire.
* You have 2 end jacks, which must be installed on your cable. If you are using a pre-made cable, with one of the ends whacked off, you only have one end to install - the crossed over end. Below are two diagrams, which show how you need to arrange the cables for each type of cable end. Decide at this point which end you are making and examine the associated picture below.

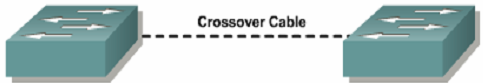
**Straight through cable:** The straight-through cable is used to connect

* Host to switch or hub
* Router to switch or hub



**Crossover cable:** The crossover cable can be used to connect

* Switch to switch
* Hub to hub
* Host to host
* Hub to switch
* Router direct to host



**Note:-** Other two pairs are used for Power on Ethernet

* PoE +VDC: 4 & 5
* PoE -VDC: 7 & 8



**Roll over cable:** Here, the connections are made in reverse order. This type of cable is used to connect the router/switch to the PC via console port for management purposes.

**Exercise 2.1 [5]**

Which type of cable will be used between switch and router?

**Exercise 2.2 [5]**

Which device is used to connect two dissimilar type of network (Use different protocol)?

# EXPERIMENT 3 – CLASSIFICATION OF IP

**Objective**

* Study and implement concepts of IP in windows.

**Time Required** : 3 hrs

**Programming Language** : NIL

**Software Required** : NIL

**Hardware Required**  :

* Computer with administrative rights

**Introduction to IP addressing**

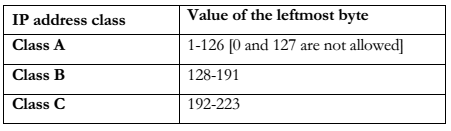
Each Network Interface Card (NIC or Network card) present in a PC is assigned one Network address called as IP address [or Network address or Logical address]. This IP address is assigned by the administrator of the network. No two PCs can have the same IP address.

There is a burned-in address on the NIC called as Physical Address [or MAC address or Hardware address]. The MAC address of a network card indicates the vendor of that card and a unique serial number.

**Rules of IPv4 addressing**

**IP address format:** IPv4 is made up of four parts, in the pattern as w. x. y. z. Each part has 8 binary bits and the values in decimal can range from 0 to 255.

**IP address classes:** IP addresses are divided into different classes. These classes determine the maximum number of hosts per network ID. Only three classes are actually used for network connectivity. The following table lists all of the address class.



**Grouping of IP addresses into different classes.**

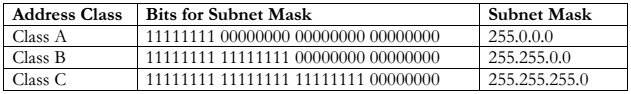
* Class A, B, C, D, E
* Class A: first bit in w is 0 and others can be anything
* 0.0.0.0 to 127.255.255.255
* ii. First bits are used for network part and the remaining for host part.
* Class B: First bit in w is1 and second bit is 0.
* 128.0.0.0 to 191.255.255.255
* ii. First 16 bits for network part and remaining host part
* Class C: first bit in w is 1, second bit in w is 1 and third bit is 0
* 192.0.0.0 to 223.255.255.255
* ii. First 24 bits for network part and last 8 bits for host part.
* Class D: first, second, third bits in w are 1 and fourth bit is 0; used for multicast.
* i. 224.0.0.0 to 247.255.255.255
* Class E: future use or experimental purposes.

**Default Subnet mask:** It is used to identify the network part from the host part. Put binary one for the parts that represent network part and zero for the part that represent host part.

* Class A: 255.0.0.0
* Class B: 255.255.0.0
* Class C: 255.255.255.0

**Note:**- We can’t have mix of 1s and 0s in subnet mask. Only consecutive 1s is followed by consecutive 0s

The following table lists the default subnet masks for each available class of TCP/IP networks.



**Exercise 3.1: [2]**

What is the IP address of your computer, and to which class it belongs?

**Exercise 3.2: [2]**

What is the subnet mask for 3.1?

**Exercise 3.3: [2]**

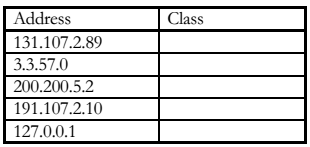
What is the Network ID for 3.1?

**Exercise 3.4: [2]**

What is the subnet mask for 3.3?

**Exercise 3.5 [2]**

Write the address class next to each IP address.



**Web Resources**

<http://www.wikihow.com/Make-a-Network-Cable>

<http://fcit.usf.edu/network/chap4/chap4.htm>

**Videos Resources**

<http://www.youtube.com/watch?v=q6wZ9vwKSEs>

<http://www.youtube.com/watch?v=bfp7oskfDXY>

<http://www.youtube.com/watch?v=iyZ9nEA_vCQ>

# EXPERIMENT 4 – CONFIGURING IP

**Objective**

* Implement concepts of IP in computer network.

**Time Required** : 3 hrs

**Programming Language** : NIL

**Software Required** : NIL

**Hardware Required**  :

* Computer with administrative rights

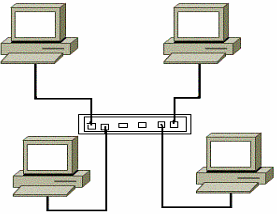
**Setting up a simple network**

In this Lab, we will learn how to make different cables and connect two PCs to create a simple Peer-to-Peer network.

**Exercise 4.1: [2]**

The two PCs will be connected with a hub between them. A hub allows for more than just two workstations to be connected depending on the number of ports on the hub. Hubs can have from 4 to 32 ports.

A simple topology is shown below in which a four node Ethernet LAN using Ethernet Hub. A UTP cable is used to connect the NIC installed inside the PC to a port on the hub.



**Tools / Preparation:** The workstations should have Network Interface Cards (NIC) installed with the proper drivers. The following resources will be required:

* Two Pentium-based workstations with a NIC in each (NIC drivers should be available and installed)
* An Ethernet switch (4 or 8 port) and two CAT5 straight-wired cables.

**Check Local Area Network Connections**

You should check the cables to verify that you have good layer 1 physical connections.

A Network Connection via Hub

**Exercise 4.2: [3]**

Check each of the two CAT 5 cables from each workstation to the hub. Verify that the pins are wired straight through by holding the two RJ-45 connectors for each cable side by side with the clip down and inspect them. All pins should have the same color wire on the same pin at both ends of the cable. (Pin 1 should match pin 1 and pin 8 should match pin 8 etc.)

**Plug in and connect the equipment**

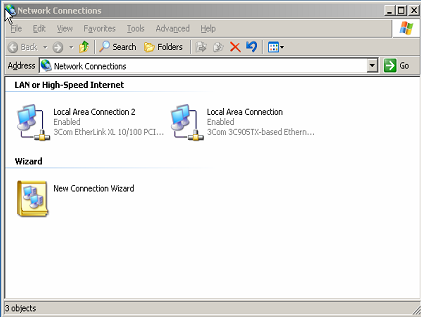
You should check the workstations and hub for exercise.

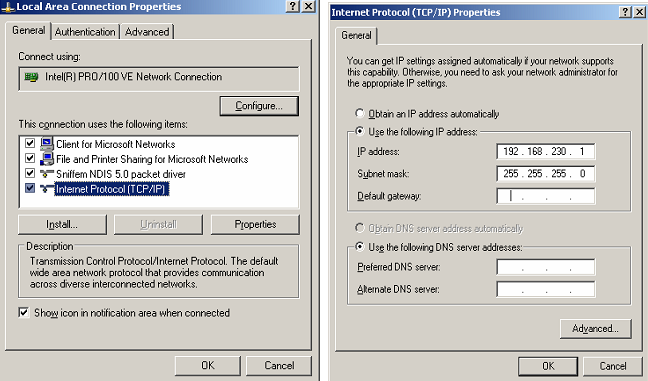
**Exercise 4.3: [5]**

Check to make sure that the NICs are installed correctly in each workstation. Plug in the workstations and turn them on. Plug the straight through cable from workstation 1 into port 1 of the hub and the cable from workstation 2 into port 2 of the hub. After the workstations have booted, check the green link light on the back of each NIC and the green lights on ports 1 and 2 of the hub to verify that the are communicating. This also verifies a good physical connection between the Hub and the NICs in the workstations (OSI Layers 1 and 2). If the link light is not on it usually indicates a bad cable connection, an incorrectly wired cable or the NIC or hub may not be functioning correctly.

**Check the TCP/IP Protocol Settings**

**Task:** Use the Control Panel/Network Connections (or Properties in Context Menu of My Network Places) to display Network Connections Window. Then use Properties in Context Menu of Local Area Connection to display Local Area Connection Properties Window. Select the TCP/IP protocol from the Configuration Tab and click on properties. Check the IP Address and Subnet mask for both workstations on the IP Address Tab.





The IP addresses can be set to anything as long as they are compatible and on the same network. Record the existing settings before making any changes in case they need to be set back (for instance, they may be DHCP clients now). For this lab, use the Class C IP network address of 192.168.230.0 and set workstation 1 to static IP address 192.168.230.1 and set workstation 2 to 192.168.230.2. Set the default subnet mask on each workstation to 255.255.0.0. For the purpose of this lab, you can leave the Gateway and DNS Server entries blank.

# EXPERIMENT 5 - BASIC NETWORKING COMMANDS IN MSDOS

**Objective**

* Study the IP configuration and packet tracing using MSDOS.

**Time Required** : 3 hrs

**Programming Language** : NIL

**Software Required** : NIL

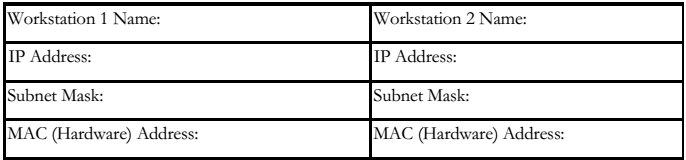
**Hardware Required**  :

* Computer with administrative rights

**Check the TCP/IP Settings with the IPCONFIG Utility**

Use the ipconfig.exe command to see your TCP/IP settings on one screen. Click on Start -> Command Prompt. Enter ipconfig /all command to see all TCP/IP related settings for your workstation.

Fill in the blanks below using the results of the IPCONFIG command from each workstation:



**Check the network connection with the Ping Utility**

Use the Ping Command to check for basic TCP/IP connectivity. Click on Start-> Command Prompt. Enter the Ping command followed by the IP address of the other workstation (Example - ping 192.168.230.1 or 192.168.230.2).

Network related commands

To know and learn about various network related commands [ping, tracert, netstat, at, net, route, arp and few definitions cum settings.

**PING Command**

Ping is a basic Internet program that lets you verify that a particular IP address exists and can accept requests. The verb ping means the act of using the ping utility or command. Ping is used diagnostically to ensure that a host computer you are trying to reach is actually operating. Various options available in the ping command:

-t repetitively sends packets.

-n number of echo to be sent

-l sending buffer size [Max: 65500 bytes]

-r count record route for count hops [3rd layer device]

**Activity 5.1**

In order to send a packet to a host [192.168.230.1] with size of 60000 bytes each. We wish to send the packets repetitively.

ping -t -l 60000 192.168.230.1

**Exercise 5.1: [2]**

Test the reach ability towards a PC [192.168.230.4].

**TRACERT Command**

If someone would like to know how he goes from his house to his office he could just tell the list of the crossroads where he passes. The same way we can ask the data sent over from your computer to the web server which way does it go, through which devices? We ask it by using the utility called trace route. In most computers today you can use this tool from the command line: In MS Windows machines it is called tracert.

Various options available in the tracert command:

-d Don’t resolve addresses to hostnames.

-h maximum\_hops Maximum number of hops to search for target

-w time-out wait timeout milliseconds for each reply.

**Activity 5.2**

To check the trace from your PC to a server

tracert 172.23.16.1

**Exercise 5.2: [2]**

Find the route from your PC to MIMS Server

**PATHPING Command**

This command is used as IP trace utility and so it is similar to the tracert command. It has some extra features compared to tracert command. It also has various options to perform.

-n Don't resolve addresses to hostnames

-h max\_hops Max number of hops to search

-p period Wait between pings (milliseconds)

-q num\_queries Number of queries per hop

-w timeout Wait timeout for each reply (milliseconds)

**NETSTAT Command**

This command is used to get information about the open connections on your system

(ports, protocols are being used, etc.), incoming and outgoing data and also the ports of remote systems to which you are connected.

Various options available in the netstat command:

-a Displays all connections and listening ports.

-e Displays Ethernet statistics. This may be combined with the -s option.

-n Displays addresses and port numbers in numerical form.

-p proto Shows connections for the protocol specified by proto; proto may be TCP or UDP. If used with the –s option to display per-protocol statistics, proto may be TCP, UDP, or IP.

-r Displays the routing table.

-s Displays per-protocol statistics. By default, statistics are shown for TCP, UDP and IP; the -p option may be used to specify a subset of the default.

**Activity 5.3**

To display all connections and listening ports

netstat –a

To find out the statistics on your Ethernet card

netstat –e

To get to know the routing table.

netstat -r

**Exercise 5.3: [2]**

Open a browser connection to http server [www.mcs.edu.pk] and write down the outcome of the command 'netstat -an'.

**ROUTE Command**

This command manipulates network routing tables. Various options available in the ROUTE command:

-f Clears the routing tables of all gateway entries. If this is used in conjunction with one of the commands, the tables are cleared prior to running the command.

Command Specifies one of four commands

PRINT Prints a route

ADD Adds a route

DELETE Deletes a route

CHANGE Modifies an existing route

Destination Specifies the host to send command.

MASK If the MASK keyword is present, the next parameter is interpreted as the netmask parameter.

Netmask If provided specifies a sub-net mask value to be associated with this route entry. If not specified, if defaults to 255.255.255.255.

Gateway Specifies gateway.

**Activity 5.4**

To display the routing table.

route PRINT

To add a route a destination

route add <destination> mask <subnetmask> <gateway> metric <number>

**Exercise 5.5: [2]**

Create a route entry in the routing table for a network 210.20.23.0 with the gateway 172.23.19.250 metric of 5.

**ARP Command**

The address resolution protocol (ARP) is a protocol used by the Internet Protocol (IP), specifically IPv4, to map IP network addresses to the hardware addresses used by a data link protocol. Various options available in the ARP command:

-a Displays current ARP entries by interrogating the current protocol data. If inet\_addr is specified, the IP and Physical addresses for only the specified computer are displayed. If more than one network interface uses ARP, entries for each ARP table are displayed.

-d Deletes the host specified by inet\_addr.

**Activity 5.5**

To display the entries in ARP cache

arp -a

To delete an ARP entry in the cache

arp –d 192.168.50.203

**Exercise 5.5: [1]**

Remove all the entries in the ARP cache and then generate a PING command to a specific PC [192.168.50.203]. Then, display all the entries in the ARP cache.

**IPCONFIG Command**

This command is used to get IP configurations present in your PC.

IPCONFIG /all Display full configuration information.

IPCONFIG /renew [adapter] Renew the IP address for the specified adapter.

IPCONFIG /flushdns Purge the DNS Resolver cache.

IPCONFIG /displaydns Display the contents of the DNS Resolver Cache. ##

**Activity 5.6**

> ipconfig ... Show information.

> ipconfig /all ... Show detailed information

> ipconfig /renew ... renew all adapters

**Exercise 5.6: [1]**

Get to know about the TCP/IP configuration on your PC using

ipconfig /all

**Web Resources**

<http://www.wikihow.com/Configure-Your-PC-to-a-Local-Area-Network>

<http://thestarman.pcministry.com/DOS/DOS7NET.htm>

**Videos Resources**

<http://www.youtube.com/watch?v=G9ixcE9Bj44>

<http://www.youtube.com/watch?v=CK_xa_avd44>

# EXPERIMENT 6 - CONFIGURING DHCP AND DNS

**Objective**

* Configure Windows 2003 as a DHCP Server
* Capture and analyze DHCP traffic generated
* Learn the structure of the Domain Name System and the role played by Name Servers.
* Configure Windows 2003 to use DNS server with various options.

**Time Required** : 3 hrs

**Programming Language** : NIL

**Software Required** : NIL

**Hardware Required** :

* Ethereal software
* WinPCap software

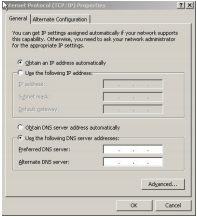
**DHCP (Dynamic Host Configuration Protocol)**

DHCP is a client/server protocol that automatically provides an IP host with its IP address and other related configuration information such as the subnet mask and default gateway. DHCP allows hosts to obtain all necessary TCP/IP configuration information from a DHCP server.

**Configure your computer**

For this lab we will make all the lab computers as hosts on their respective network. Thus at every computer modify the network configurations as follows:

Setup the first computer in every network as a DHCP server and have the other computers in the group point to it as DHCP clients. Thus computer 192.168.1.1 –which will be configured as DHCP server has static IP but all the clients get IP address from the server.

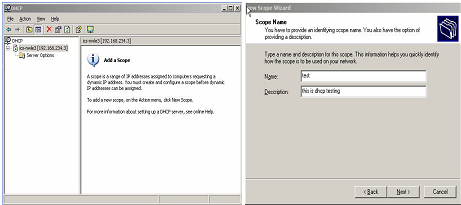
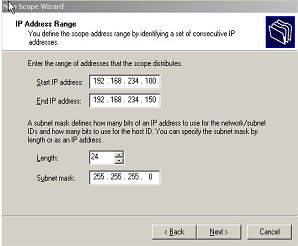


**Configure Windows 2003 as a DHCP Client**

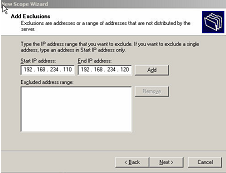
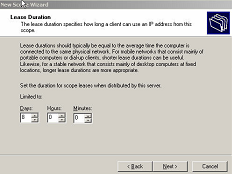
Right-click on “My Network Places” on desktop and select properties. Select any one of the local area connections and click. Click Properties. Local Area connection properties window appears. Select Internet Protocol (TCP/IP) and click Properties. Internet Protocol (TCP/IP) Properties window appears. Select the radio button ‘obtain an IP address automatically’.

**Configure Windows 2003 as a DHCP Server**

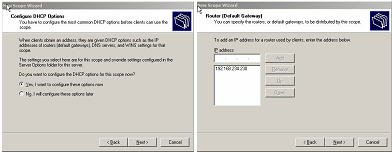
Open Administrative Tools from start menu and select DHCP. DHCP Manager appears. Click on the computer and right-click and select New Scope. New Scope Wizard appears. Enter the name of the scope and its description. Enter the starting and ending IP address of the scope as instructed by the instructor.

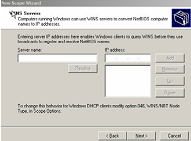
Click Next. If needed, add exclusion range and click Add. Click Next. On the lease duration, click Next unless specified by the instructor.

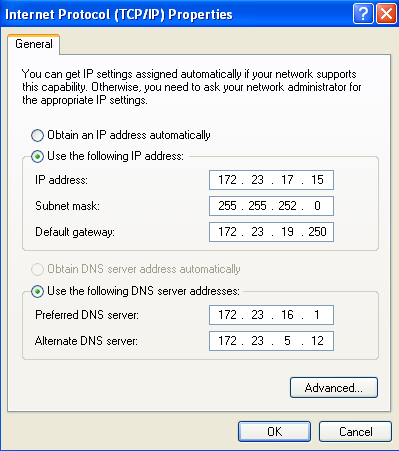
 

Select Yes for DHCP configure options and Click Next. If needed specify the router [default gateway] address and click Add. Click Next.



Click Next [for DNS server]. Click Next [for WINS server]. Select Yes for activating the scope. Click Next. Completing the new scope wizard appears. Click Finish. DHCP window appears.

**DNS (DOMAIN NAME SYSTEM)**

In the context of DNS, A Name Server is the application that is acting as the server for the DNS protocol. A Name Server performs two primary tasks

* Maintains among other things the host-name to IP address mappings for the hosts in its zone.
* Responds to DNS queries. Recall that a query is basically a partial resource record. The name server job is to return the corresponding matching resource records.

**Configure Windows 2003 to use DNS**

Open the network connection properties and Click Local Area Connection Properties. Select Internet Protocol (TCP/IP) and click Properties. TCP/IP Window appears and set preferred DNS Server to 172.23.16.1 and Alternate DNS Server is 172.23.5.12.

**Exercise 6.1: [2]**

Why is that the Source IP address of the DHCP Discover all 0s?

**Exercise 6.2: [2]**

Why is that the Destination IP address of the DHCP Discover all 1s?

**Exercise 6.3: [2]**

What is the use of physical address in DHCP?

**Exercise 6.4: [2]**

Why is the Destination IP address of DHCP Offer all 1s?

**Exercise 6.5: [2]**

What is the IP address of DNS server in the Lab?

**Web Resources**

<http://www.ucs.cam.ac.uk/support/windows-support/winsuptech/activedir/dnsconfig>

<http://support.microsoft.com/kb/323416>

**Videos Resources**

<http://www.youtube.com/watch?v=fwdr761s5U8>

<http://www.youtube.com/watch?v=kwnKoQqI2Cs>

# EXPERIMENT 7 - SUBNETTING

**Objective**

* Introduction to subnetting concepts using class C and its implementation in LAB.

**Time Required** : 3 hrs

**Programming Language** : NIL

**Software Required** : NIL

**Hardware Required**  : PC with administrative access

## Understand IP Addresses

An IP address is an address used in order to uniquely identify a device on an IP network. The address is made up of 32 binary bits, which can be divisible into a network portion and host portion with the help of a subnet mask. The 32 binary bits are broken into four octets (1 octet = 8 bits). Each octet is converted to decimal and separated by a period (dot). For this reason, an IP address is said to be expressed in dotted decimal format (for example, 172.16.81.100). The value in each octet ranges from 0 to 255 decimal, or 00000000 - 11111111 binary.

Here is how binary octets convert to decimal: The right most bit, or least significant bit, of an octet holds a value of 20. The bit just to the left of that holds a value of 21. This continues until the left-most bit, or most significant bit, which holds a value of 27. So if all binary bits are a one, the decimal equivalent would be 255 as shown here:

1 1 1 1 1 1 1 1

128 64 32 16 8 4 2 1 (128+64+32+16+8+4+2+1=255)

Here is a sample octet conversion when not all of the bits are set to 1.

0 1 0 0 0 0 0 1

0 64 0 0 0 0 0 1 (0+64+0+0+0+0+0+1=65)

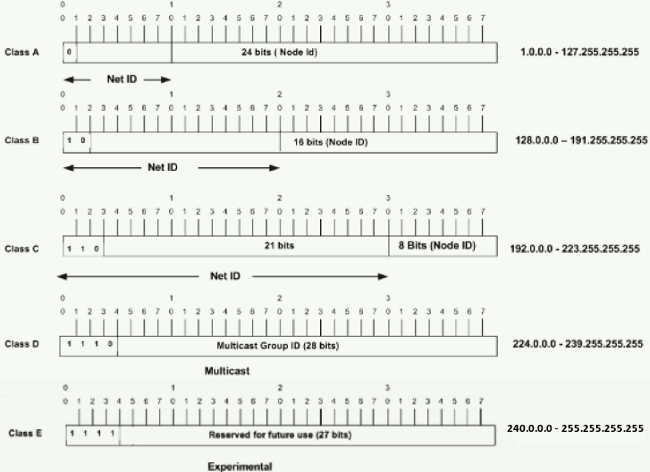
And this sample shows an IP address represented in both binary and decimal.

10. 1. 23. 19 (decimal)

00001010.00000001.00010111.00010011 (binary)

These octets are broken down to provide an addressing scheme that can accommodate large and small networks. There are five different classes of networks, A to E. This document focuses on classes A to C

Given an IP address, its class can be determined from the three high-order bits (the three left-most bits in the first octet). [Figure 1](http://www.cisco.com/c/en/us/support/docs/ip/routing-information-protocol-rip/13788-3.html" \l "figone) shows the significance in the three high order bits and the range of addresses that fall into each class. For informational purposes, Class D and Class E addresses are also shown.

[](http://www.cisco.com/c/dam/en/us/support/docs/ip/routing-information-protocol-rip/13788-3-00.gif)

In a Class A address, the first octet is the network portion, so the Class A example in [Figure 1](http://www.cisco.com/c/en/us/support/docs/ip/routing-information-protocol-rip/13788-3.html" \l "figone) has a major network address of 1.0.0.0 - 127.255.255.255. Octets 2, 3, and 4 (the next 24 bits) are for the network manager to divide into subnets and hosts as he/she sees fit. Class A addresses are used for networks that have more than 65,536 hosts (actually, up to 16777214 hosts!).

In a Class B address, the first two octets are the network portion, so the Class B example in [Figure 1](http://www.cisco.com/c/en/us/support/docs/ip/routing-information-protocol-rip/13788-3.html" \l "figone) has a major network address of 128.0.0.0 - 191.255.255.255. Octets 3 and 4 (16 bits) are for local subnets and hosts. Class B addresses are used for networks that have between 256 and 65534 hosts.

In a Class C address, the first three octets are the network portion. The Class C example in [Figure 1](http://www.cisco.com/c/en/us/support/docs/ip/routing-information-protocol-rip/13788-3.html" \l "figone) has a major network address of 192.0.0.0 - 223.255.255.255. Octet 4 (8 bits) is for local subnets and hosts - perfect for networks with less than 254 hosts.

## Network Masks

A network mask helps you know which portion of the address identifies the network and which portion of the address identifies the node. Class A, B, and C networks have default masks, also known as natural masks, as shown here:

Class A: 255.0.0.0

Class B: 255.255.0.0

Class C: 255.255.255.0

An IP address on a Class A network that has not been subnetted would have an address/mask pair similar to: 8.20.15.1 255.0.0.0. In order to see how the mask helps you identify the network and node parts of the address, convert the address and mask to binary numbers.

8.20.15.1 = 00001000.00010100.00001111.00000001

255.0.0.0 = 11111111.00000000.00000000.00000000

Once you have the address and the mask represented in binary, then identification of the network and host ID is easier. Any address bits which have corresponding mask bits set to 1 represent the network ID. Any address bits that have corresponding mask bits set to 0 represent the node ID.

8.20.15.1 = 00001000.00010100.00001111.00000001

255.0.0.0 = 11111111.00000000.00000000.00000000

-----------------------------------

net id | host id

netid = 00001000 = 8

hostid = 00010100.00001111.00000001 = 20.15.1

## Understand Subnetting

Subnetting allows you to create multiple logical networks that exist within a single Class A, B, or C network. If you do not subnet, you are only able to use one network from your Class A, B, or C network, which is unrealistic.

Each data link on a network must have a unique network ID, with every node on that link being a member of the same network. If you break a major network (Class A, B, or C) into smaller subnetworks, it allows you to create a network of interconnecting subnetworks. Each data link on this network would then have a unique network/subnetwork ID. Any device, or gateway, that connects n networks/subnetworks has n distinct IP addresses, one for each network / subnetwork that it interconnects.

In order to subnet a network, extend the natural mask with some of the bits from the host ID portion of the address in order to create a subnetwork ID. For example, given a Class C network of 204.17.5.0 which has a natural mask of 255.255.255.0, you can create subnets in this manner:

204.17.5.0 - 11001100.00010001.00000101.00000000

255.255.255.224 - 11111111.11111111.11111111.11100000

--------------------------|sub|----

By extending the mask to be 255.255.255.224, you have taken three bits (indicated by "sub") from the original host portion of the address and used them to make subnets. With these three bits, it is possible to create eight subnets. With the remaining five host ID bits, each subnet can have up to 32 host addresses, 30 of which can actually be assigned to a device since host ids of all zeros or all ones are not allowed (it is very important to remember this). So, with this in mind, these subnets have been created.

204.17.5.0 255.255.255.224 host address range 1 to 30

204.17.5.32 255.255.255.224 host address range 33 to 62

204.17.5.64 255.255.255.224 host address range 65 to 94

204.17.5.96 255.255.255.224 host address range 97 to 126

204.17.5.128 255.255.255.224 host address range 129 to 158

204.17.5.160 255.255.255.224 host address range 161 to 190

204.17.5.192 255.255.255.224 host address range 193 to 222

204.17.5.224 255.255.255.224 host address range 225 to 254

There are two ways to denote these masks. First, since you use three bits more than the "natural" Class C mask, you can denote these addresses as having a 3-bit subnet mask. Or, secondly, the mask of 255.255.255.224 can also be denoted as /27 as there are 27 bits that are set in the mask. This second method is used with [CIDR](http://www.cisco.com/c/en/us/support/docs/ip/routing-information-protocol-rip/13788-3.html" \l "cidr). With this method, one of these networks can be described with the notation prefix/length. For example, 204.17.5.32/27 denotes the network 204.17.5.32 255.255.255.224. When appropriate, the prefix/length notation is used to denote the mask throughout the rest of this document.

**Exercise 7.1: [10]**

Consider the following Class C Network ID

192.168.1.0

How many bit should be borrowed if maximum of hosts in each subnet is 50?

Perform the complete subnetting exercise showing Network IDs, Broadcast IDs and IP address range in each subnet.

# EXPERIMENT 8 – SUBNETTING CLASS A AND B

**Objective**

* Subnetting concepts using Class b and Class A and its implementation in LAB.

**Time Required** : 3 hrs

**Programming Language** : NIL

**Software Required** : NIL

**Hardware Required**  : PC with administrative access

### **Class B Subnetting**

The first two octets of a [Class B network](http://www.omnisecu.com/tcpip/internet-layer-ip-addresses.php" \t "_blank) is used to represent the network and the last two octets are used to represent the host. The default format for a [Class B IPv4 address](http://www.omnisecu.com/tcpip/internet-layer-ip-addresses.php" \t "_blank) is Network.Network.Host.Host.

Let us consider an example of [Class B network](http://www.omnisecu.com/tcpip/internet-layer-ip-addresses.php" \t "_blank) 172.16.0.0 - 255.255.0.0. The [binary representation](http://www.omnisecu.com/tcpip/binary-decimal-hexadecimal-numbers-and-conversions.php" \t "_blank) of the above network and [subnet mask](http://www.omnisecu.com/tcpip/internet-layer-ip-addresses.php" \t "_blank) is

|  |  |  |
| --- | --- | --- |
| **Component** | **[Binary](http://www.omnisecu.com/tcpip/binary-decimal-hexadecimal-numbers-and-conversions.php" \t "_blank)** | **Decimal** |
| Address Part | 10101100.00010000.00000000.00000000 | 172.16.0.0 |
| SN Mask | 11111111.11111111.00000000.00000000 | 255.255.0.0 |

If all the bits in the host part are "0", that represents the network id.

If all the bits in the host part are "0" except the last bit, it is the first usable IPv4 address.

If all the bits in the host part are "1" except the last bit, it is the last usable IPv4 address.

If all the bits in the host part are "1", that represents the [directed broadcast address](http://www.omnisecu.com/tcpip/internet-layer-ip-addresses.php" \t "_blank).

All the [IPv4 addresses](http://www.omnisecu.com/tcpip/internet-layer-ip-addresses.php" \t "_blank) between the first and last [IPv4 addresses](http://www.omnisecu.com/tcpip/internet-layer-ip-addresses.php" \t "_blank) (including the first and last) can be used to configure the devices.

#### **Class B - One Bit Subnetting**

If we include one bit from the host part to the network part, the [subnet mask](http://www.omnisecu.com/tcpip/internet-layer-ip-addresses.php" \t "_blank) is changed into 255.255.128.0 The single bit can have two values in third octet, either 0 or 1.

10101100.00010000.**0** | 0000000.00000000  
11111111.11111111.**1** | 0000000.00000000

That means, we can get two subnets if we do a single bit subnetting.

|  |  |  |  |
| --- | --- | --- | --- |
| SN No | Description | Binaries | Decimal |
| 1 | Network Address | 10101100.00010000.00000000.00000000 | 172.16.0.0 |
| First IPv4 address | 10101100.00010000.00000000.00000001 | 172.16.0.1 |
| Last IPv4 address | 10101100.00010000.01111111.11111110 | 172.16.127.254 |
| Broadcast Address | 10101100.00010000.01111111.11111111 | 172.16.127.255 |
| 2 | Network Address | 10101100.00010000.10000000.00000000 | 172.16.128.0 |
| First IPv4 address | 10101100.00010000.10000000.00000001 | 172.16.128.1 |
| Last IPv4 address | 10101100.00010000.11111111.11111110 | 172.16.255.254 |
| Broadcast Address | 10101100.00010000.11111111.11111111 | 172.16.255.255 |

The network 172.16.0.0 is divided into two networks, each network has 32768 total IPv4 addresses and 32766 usable [IPv4 addresses](http://www.omnisecu.com/tcpip/internet-layer-ip-addresses.php" \t "_blank) (two [IPv4 addresses](http://www.omnisecu.com/tcpip/internet-layer-ip-addresses.php" \t "_blank) are used in each subnet to represent the [network address](http://www.omnisecu.com/tcpip/internet-layer-ip-addresses.php" \t "_blank) and the [directed broadcast address](http://www.omnisecu.com/tcpip/internet-layer-ip-addresses.php" \t "_blank)). The [subnet mask](http://www.omnisecu.com/tcpip/internet-layer-ip-addresses.php" \t "_blank) for one bit subnetting is 255.255.128.0.

#### **Class B - Two Bit Subnetting**

If we include two bits from the host part to the network part, the [subnet mask](http://www.omnisecu.com/tcpip/internet-layer-ip-addresses.php" \t "_blank) is changed into 255.255.192.0. The two bits added to network part can have four possible values in third octet, 00, 01, 10, and 11.

10101100.00010000.**00** | 000000.00000000  
11111111.11111111.**11** | 000000.00000000

That means, we can get four networks if we do a two bit subnetting.

|  |  |  |  |
| --- | --- | --- | --- |
| SN No | Description | Binaries | Decimal |
| 1 | Network Address | 10101100.00010000.00000000.00000000 | 172.16.0.0 |
| First IPv4 address | 10101100.00010000.00000000.00000001 | 172.16.0.1 |
| Last IPv4 address | 10101100.00010000.00111111.11111110 | 172.16.63.254 |
| Broadcast Address | 10101100.00010000.00111111.11111111 | 172.16.63.255 |
| 2 | Network Address | 10101100.00010000.01000000.00000000 | 172.16.64.0 |
| First IPv4 address | 10101100.00010000.01000000.00000001 | 172.16.64.1 |
| Last IPv4 address | 10101100.00010000.01111111.11111110 | 172.16.127.254 |
| Broadcast Address | 10101100.00010000.01111111.11111111 | 172.16.127.255 |
| 3 | Network Address | 10101100.00010000.10000000.00000000 | 172.16.128.0 |
| First IPv4 address | 10101100.00010000.10000000.00000001 | 172.16.128.1 |
| Last IPv4 address | 10101100.00010000.10111111.11111110 | 172.16.191.254 |
| Broadcast Address | 10101100.00010000.10111111.11111111 | 172.16.191.255 |
| 4 | Network Address | 10101100.00010000.11000000.00000000 | 172.16.192.0 |
| First IPv4 address | 10101100.00010000.11000000.00000001 | 172.16.192.1 |
| Last IPv4 address | 10101100.00010000.11111111.11111110 | 172.16.255.254 |
| Broadcast Address | 10101100.00010000.11111111.11111111 | 172.16.255.255 |

The network 172.16.0.0 is divided into four networks, each network has 16384 total [IPv4 addresses](http://www.omnisecu.com/tcpip/internet-layer-ip-addresses.php" \t "_blank) and 16382 usable [IPv4 addresses](http://www.omnisecu.com/tcpip/internet-layer-ip-addresses.php" \t "_blank) (two [IPv4 addresses](http://www.omnisecu.com/tcpip/internet-layer-ip-addresses.php" \t "_blank) are used in each subnet to represent the [network address](http://www.omnisecu.com/tcpip/internet-layer-ip-addresses.php" \t "_blank) and the [directed broadcast address](http://www.omnisecu.com/tcpip/internet-layer-ip-addresses.php" \t "_blank)). The [subnet mask](http://www.omnisecu.com/tcpip/internet-layer-ip-addresses.php" \t "_blank) for one bit subnetting is 255.255.192.0.

**Exercise 8.1: (LAB 8) [5]**

Consider the following IP address and subnet mask.

131.0.10.11

255.255.128.0

* + 1. What is the Network ID for the above IP address?
    2. How many bit have been borrowed in this case?
    3. What is the broadcast ID for the Network ID you found earlier?

**Exercise 8.2: (LAB 8) [5]**

Implement the concepts of subnetting in LAB for Class A network ID.

# EXPERIMENT 9 – PACKET TRACER AND DESIGNING TOPOLOGY

**Objective**

* Introduction to Packet Tracer interface.
* Learn how to use existing topologies and build your own.

**Time Required** : 3 hrs

**Programming Language** : NIL

**Software Required** : Packet Tracer

**Hardware Required** : NIL

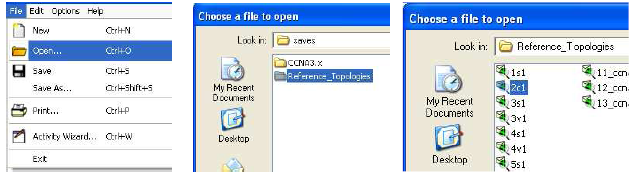
**Introduction to Packet Tracer**

**What is Packet Tracer?** Packet Tracer is a protocol simulator developed by Dennis Frezzo and his team at Cisco Systems. Packet Tracer (PT) is a powerful and dynamic tool that displays the various protocols used in networking, in either Real Time or Simulation mode. This includes layer 2 protocols such as Ethernet and PPP, layer 3 protocols such as IP, ICMP, and ARP, and layer 4 protocols such as TCP and UDP. Routing protocols can also be traced.

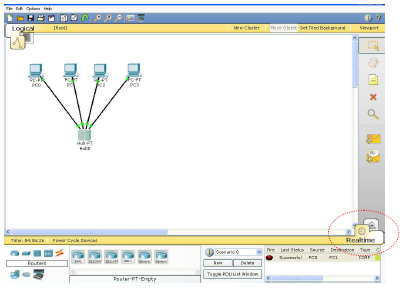
**Introduction to the Packet Tracer Interface using a Hub Topology**

**Step 1:** Start Packet Tracer and Entering Simulation Mode and Launch Packet Tracer program from the program list.

**Step 2:** Open an existing topology and Perform the following steps to open the **2c1** topology.



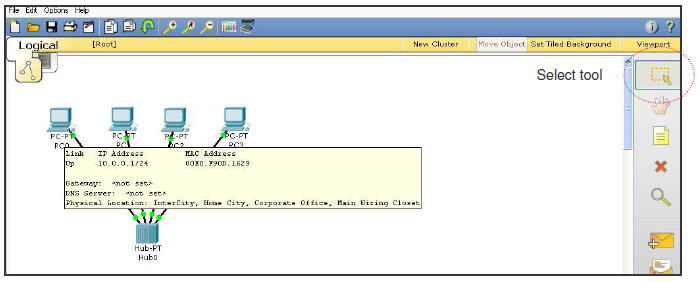
By default, the topology opens in Realtime mode.



We will examine the difference between Real-time and Simulation modes in a moment.

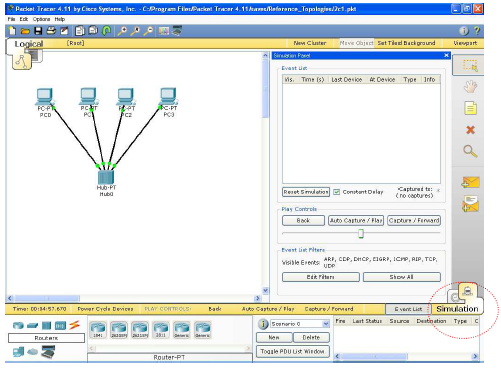
To view the IP address, subnet mask, default gateway, and MAC address of a host, move the cursor over that computer.

Be sure the Select box is checked at the top of the tool box. Viewing PC0 information using the Select tool:



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”.



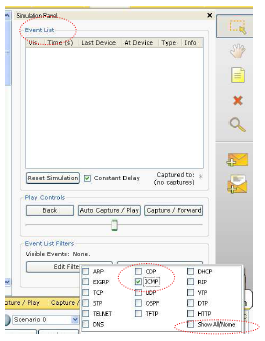
**Step 3:** PC0 pinging PC1

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.

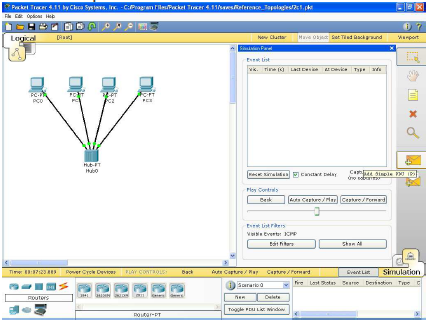
When the user issues the ping command, most operating systems send multiple (four or five) ICMP Echo messages. When the destination device receives the ping, Echo Request, it issues an Echo Reply.

**Command issued from PC0: ping 10.0.0.2**

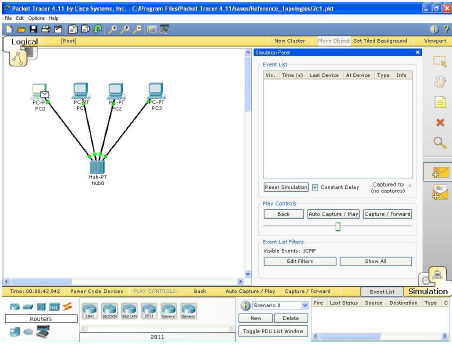
Packet Tracer allows us to either issue the command from the command prompt or to use the Add Simple PDU tool. We will look at both ways to do this. In order to view only the “pings”, in the Event List Filter, click on SHOW ALL/NONE to clear all protocols, and then click on ICMP to select only that protocol.



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).

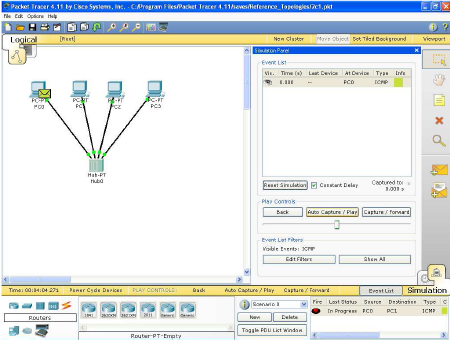


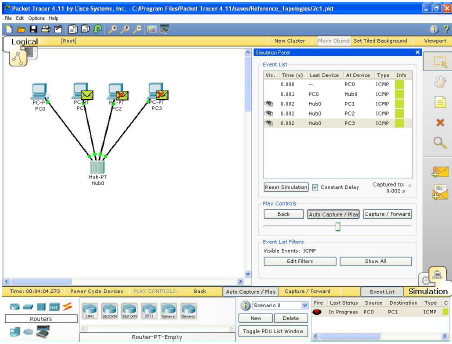
By clicking on the Auto Capture/Play button, this will capture all events in interval of 0.001 second. For example, the first event is the building of the ICMP packet and encapsulating it in an Ethernet frame. The next event will send this Ethernet frame from the Ethernet NIC in PC0 to the Hub.

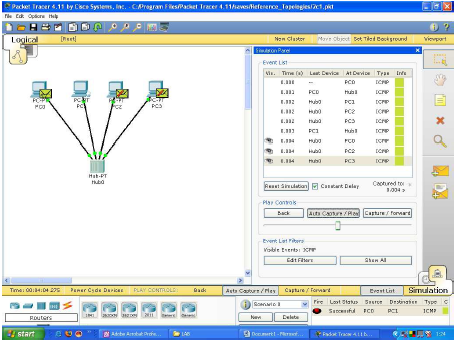
Notice that the hub floods all of the frames out all ports except the port incoming port.

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.



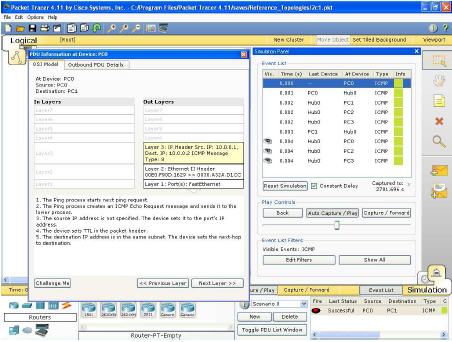




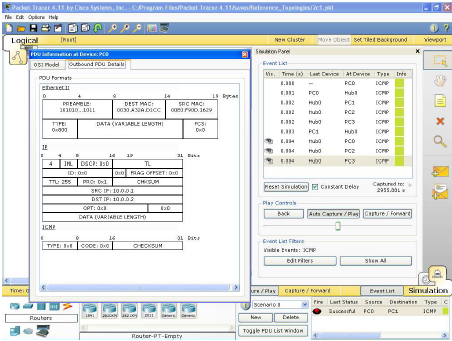
**Step 4:** Viewing the frame (Protocol Analyzer)

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 second 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. If you only see the IP packet and the ICMP message, and do not see the Ethernet II frame, click on the next ICMP Info box. This happened because we are looking at the IP packet before it got encapsulated into an Ethernet frame.



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.



**Exercise 9.1: [10]**

Build two topologies separately having four PC i.e. PC1, PC2, PC3 and PC4 connected to central location (switch and hub). Send 5, 10, 15, 20 and 25 packet from PC1 to PC4 using **ping** command in each topology.

**OUTPUTS:** Draw the graph showing the performance of both topologies between no packet and total time taken in each session.

**Web Resources**

<http://www.cisco.com/web/learning/netacad/course_catalog/docs/Cisco_PacketTracer_DS.pdf>

**Videos Resources**

<http://www.packettracernetwork.com/tutorials/video-tutorials.html>

<http://www.youtube.com/watch?v=hrRT2UxS_L8>

<http://engweb.info/cisco/Packet%20Tracer%20Tutorials.html>

<http://www.youtube.com/watch?v=VqMeJ-WH4E0>

# EXPERIMENT 10 – ROUTER CONFIGURATION

**Objective**

* Learn how to configure routers.

**Time Required :** 3 hrs

**Programming Language :** Nil

**Software Required :** Router Simulator (Boson NetSim or Packet Tracer)

**Hardware Required :** NIL

**Steps:**

Switch ON the router (if new router that is not configured it will ask -----

Would u like to enter initial configuration dialog[yes/no]: no

Press return to get started (enter)

Router>

**User Mode/User Executable Mode**

Router> enable (enter)

Router#

**Privileged Mode/Enable Mode – Executable Mode.**

The following commands can be executed in this mode

Router#show running-config(enter)

Router#debug xxx

Router#copy xxx

Router#configure terminal(enter)

Router(config)#

**Global Configuration Mode - Any configuration change in this mode affects the whole router.**

Router(config)#interface e 0/fastethernet 0/ S0 / S 1(enter)

Router(config-if)#

**Specific Configuration Mode – configuration changes to specific part of the router like lines and interfaces.**

**Setting User mode Password**

Router(config)#Line console 0(enter)

Router(config-line)#password xxxx

Router(config-line)#login

**To set username & password for the user mode**

Router(config)#username xxxx password xxxx

Router(config)#Line console 0

Router(config-line)#login local

**To change the hostname**

Router(config)#hostname HOR(enter)

HOR(config)#

**To encrypt all the passwords**

Router(config)#service password-encryption

**To set password for the privileged mode**

Router(config)#enable password/secret xxxx

**Exercise 10.1.**  . What are the different modes in a router? **[2.5]**

**Exercise 10.2.** What are the commands to encrypt our passwords? **[2.5]**

**Exercise 10.3.** Write commands to set password to the privilege mode? **[2.5]**

**Exercise 10.4.** Write commands to set username and password to the user mode? **[2.5]**

# EXPERIMENT 11 – STATIC ROUTING

**Objective**

* Learn how to configure static routing in routers.

**Time Required :** 3 hrs

**Programming Language :** Nil

**Software Required :** Router Simulator (Boson NetSim or Packet Tracer)

**Hardware Required :** NIL

**STATIC ROUTING**

Static routing is a form of [routing](https://en.wikipedia.org/wiki/Routing" \o "Routing) that occurs when a router uses a manually-configured routing entry, rather than information from a dynamic routing traffic. In many cases, static routes are manually configured by a [network administrator](https://en.wikipedia.org/wiki/Network_administrator" \o "Network administrator) by adding in entries into a [routing table](https://en.wikipedia.org/wiki/Routing_table" \o "Routing table), though this may not always be the case.

****

****

\* **All interfaces are administratively down when the router is switched on. We change their status to up by using the command ‘no shutdown’.**

\*\* **In static routing, we are manually adding the destination network to our Routing table.**

**Router(config-if)# ip route <dest. N/W> <DSNM> <next hop addr>**

**Next hop address** refers to the address of the next router that receives the packet and then forwards it to the remote location.

**Commands**

**1** . **Router**#show running-config

This will display the current configuration of the router.

**2** . **Router**#show controllers serial 0

To identify the DCE & DTE ends of the Serial cable.

**3** . **Router**#show interface ethernet 0

This will displays the details of ethernet interface.

**4** . **Router**#show interface serial 0

This will display the details of serial interface.

**5**. **Router**#show ip interface brief

This will display the interface & line protocol status in a tabular format.

**Outputs**

**Router#sh running-config**

interface Ethernet0

ip address 10.0.0.1 255.0.0.0

!

interface Ethernet1

no ip address

shutdown

!

interface Serial0

ip address 20.0.0.2 255.0.0.0

clockrate 64000

!

interface Serial1

no ip address

shutdown

!

IP route 30.0.0.0 255.0.0.0 20.0.0.1

!

line con 0

line aux 0

line vty 0 4

login

!

End

**Router#sh int e0**

Ethernet0 is up, line protocol is up

Hardware is Lance, address is 0010.7b80.c3c6 (bia 0010.7b80.c3c6)

Internet address is 10.0.0.1/8

MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec,

reliability 255/255, txload 1/255, rxload 1/255

Encapsulation ARPA, loopback not set, keepalive set (10 sec)

**Router#sh int s0**

Serial0 is up, line protocol is up

Hardware is HD64570

Internet address is 20.0.0.2/8

MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,

reliability 255/255, txload 1/255, rxload 1/255

Encapsulation HDLC, loopback not set, keep alive set (10 sec)

**Router#show ip int brief**

**State 1 -** When both **no shutdown** and **clock rate** is applied to corresponding interfaces

**Interface IP-Address OK? Method Status Protocol**

Ethernet0 30.0.0.1 YES manual **up up**

Serial0 20.0.0.2 YES manual **up up**

**Router#show ip int brief**

**State 2** - When **clock rate** is not given on DCE end & layer 1 problem

**Interface IP-Address OK? Method Status Protocol**

Ethernet0 30.0.0.1 YES manual **up up**

Serial0 20.0.0.2 YES manual **up down**

**Router#show ip int brief**

**State 3** - When the other end serial interface is **shut down**

**Interface IP-Address OK? Method Status Protocol**

Ethernet0 30.0.0.1 YES manual **up up**

Serial0 20.0.0.2 YES manual **down down**

**Router#show controllers s 0**

HD unit 0, idb = 0xB883C, driver structure at 0xBDB98

buffer size 1524 HD unit 0, **V.35 DCE cable, clockrate 64000**

**Router#show ip route**

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, \* - candidate default

Gateway of last resort is not set

C 20.0.0.0/8 is directly connected, Serial0

C 10.0.0.0/8 is directly connected, Ethernet0

S 30.0.0.0/8 [1/0] via 20.0.0.1

**Router#sh protocols**

Global values:

Internet Protocol routing is enabled

Ethernet0 is up, line protocol is up

Internet address is 10.0.0.1/8

Ethernet1 is administratively down, line protocol is down

Serial0 is up, line protocol is up

Internet address is 20.0.0.2/8

Serial1 is administratively down, line protocol is down

**Difference between**

**Request timed out**

**Destination host unreachable**

**Reply from <ip add> : <byte= > time<xms TTL=xxx**

**Request timed out**

When the packet is lost in transition, we will get **“request timed out”** message.

**Destination host unreachable**

If the host doesn’t know the route to the destination **- “Destination unreachable”** message is displayed i.e., the specified address is not present in the routing table

**Reply from <ip add> : <byte= > time<xms TTL=xxx**

Reply from the destination indicates that the connection exists

**Time To Live (TTL) –** a field in an IP header that indicates the no. of routers (hops) the packet can cross. TTL for systems is 128 and for routers it is 255. If the destination is not reached before the TTL expires, then the packet is dropped. This stops IP packets from continuously circling around in the network looking for a home.

**Exercise 11.1.** Implement the concept of static routing for the given diagram in LAB 11.  **[10]**

# EXPERIMENT 12 – DYNAMIC ROUTING

**Objective**

* Learn how to configure dynamic routing protocol in routers.

**Time Required :** 3 hrs

**Programming Language :** Nil

**Software Required :** Router Simulator (Boson NetSim or Packet Tracer)

**Hardware Required :** NIL

Dynamic routing is a networking technique that provides optimal data routing. Unlike static routing, dynamic routing enables routers to select paths according to real-time logical network layout changes. In dynamic routing, the routing protocol operating on the router is responsible for the creation, maintenance and updating of the dynamic routing table. In static routing, all these jobs are manually done by the system administrator.  
  
Dynamic routing uses multiple algorithms and protocols. The most popular are Routing Information Protocol (RIP), Interior Gateway Routing Protocol (IGRP) () and Open Shortest Path First (OSPF).

The cost of routing is a critical factor for all organizations. The least-expensive routing technology is provided by dynamic routing, which automates table changes and provides the best paths for data transmission.  
  
Typically, dynamic routing protocol operations can be explained as follows:

1. The router delivers and receives the routing messages on the router interfaces.
2. The routing messages and information are shared with other routers, which use exactly the same routing protocol.
3. Routers swap the routing information to discover data about remote networks.
4. Whenever a router finds a change in topology, the routing protocol advertises this topology change to other routers.





In **RIP**, we specify only those networks that belong to us. RIP sends routing table updates to its neighbors for every 30secs. RIP uses hop count as a unit of metric. The administrative distance of RIP is 120

**IGRP** uses autonomous number system. Here, only the networks that come under the same autonomous system number will communicate with each other. Autonomous number is provided by ISP. (By default, networks in different Autonomous system will not communicate, for different Autonomous systems to communicate **redistribution** should be done - **CCNP concept**).

IGRP sends updates for every 90secs and uses bandwidth and delay as unit of metric. IGRP has an administrative distance of 100

|  |  |  |  |
| --- | --- | --- | --- |
| **Time Intervals** | | **RIP** | **IGRP** |
| Update Interval | | 30 | 90 |
| Hold-down timer | | 180 | 280 |
| Invalid after | | 180 | 270 |
| Flushed after | 240 | | |

1 . **Router#**debug ip rip

It shows the updates sent to the neighbor routers for every 30 sec.

2. **Router#**Clear ip route \*

This allows the routing table to switch to the new updates by clearing the old entries.

3 . **Router#**debug ip igrp transactions

Displays the routing table updates that is sent for every 90sec

4 . **Router#**Undebug all (u all)

To stop all debug commands those are active.

5 . **Router#**debug ip routing

Displays the dynamic changes made in the routing table

**Exercise 12.1.** Implement the concept of static routing for the given diagram in LAB 11. **[10]**

# EXPERIMENT 13 – ACCESS CONTROL LIST IN ROUTER

**Objective**

* Learn how to configure dynamic routing protocol in routers.

**Time Required :** 3 hrs

**Programming Language :** Nil

**Software Required :** Router Simulator (Boson NetSim or Packet Tracer)

**Hardware Required :** NIL

**Access Control Lists (ACLs)**

Access Control Lists (ACLs) allow a router to permit or deny packets based on a variety of criteria. The ACL is configured in global mode, but is applied at the interface level. An ACL does not take effect until it is expressly applied to an interface with the ip access-group command. Packets can be filtered as they enter or exit an interface.

If a packet enters or exits an interface with an ACL applied, the packet is compared against the criteria of the ACL. If the packet matches the first line of the ACL, the appropriate “permit” or “deny” action is taken. If there is no match, the second line’s criterion is examined. Again, if there is a match, the appropriate action is taken; if there is no match, the third line of the ACL is compared to the packet.

This process continues until a match is found, at which time the ACL stops running. If no match is found, a default “deny” takes place, and the packet will not be processed. When an ACL is configured, if a packet is not expressly permitted, it will be subject to the implicit deny at the end of every ACL. This is the default behaviour of an ACL and cannot be changed.

A standard ACL is concerned with only one factor, the source IP address of the packet. The destination is not considered.



**STANDARD ACL** - Access-list No. 1-99

Action, ACL number, Source IP, SWCM (Source Wild Card Mask) are the parameters to be considered while configuring standard ACL which considers only the source address when rules are defined and to be checked.

**R1(config)#Access-list <Al No> <Action> <SIP> <SWCM>**

**R1(config)#**Access-list 5 deny 30.0.0.10 0.0.0.0

**R1(config)#**Access-list 5 permit 30.0.0.15 0.0.0.0 - *Here it considers the source address only*

Once you have defined the access-list, binding should be done at the interface required (E0 or S0).

For an interface at any time there are maximum of two binding (in and out).

**R1(config)#**int e 0 / s0

**R1(config-if)#**ip Access-group 5 in/out

For binding we consider two directions: In & Out. This specifies the direction in which the data packets are moving with respect to the Router.

**Extended ACL** - Access-list No. 100-199

Action, ACL number, Source IP, SWCM (Source Wild Card Mask), Destination IP, DWCM, Protocol, interface and the destination port number are the parameters considered while configuring Extended ACL.

**R1(config)#Access-list<AlNo><Action><protocol><SIP><SWCM><DIP><DWCM>eq <PORT No./Name>**

*Port no - can be replaced by the port names.*

*Source IP – From IP address.*

*Destination IP – To IP address.*

*Eq – equivalent to.*

*Action – permit or deny.*

**R1(config)#**Access-list 101 deny TCP 10.0.0.10 0.0.0.0 30.0.0.10 0.0.0.0 eq 80

**R1(config)#**Access-list 101 permit TCP 10.0.0.10 0.0.0.0 30.0.0.10 0.0.0.0 eq 23

**R1(config)#**Access-list 101 deny ICMP any any

**R1(config)#Access-list<AlNo><Action><protocol><SIP><SWCM><DIP><DWCM>eq <PORT No./Name>log**

**R1(config)#**Access-list 101 deny TCP 10.0.0.10 0.0.0.0 30.0.0.10 0.0.0.0 eq ftp log

*Log - gives information about the port no of source, number of packets send, number of matches made etc.,*

**NAMED ACCESS LIST**

**Standard**

**R1(config)#IP Access-list standard <Name>**

**R1(config)**#IP access-list standard moon

**R1(config-std-nacl)# <Action> <SIP> <SWCM>**

**R1(config-std-nacl)**#deny 30.0.0.10 0.0.0.0

**R1(config-std-nacl)#** permit 30.0.0.15 0.0.0.0

**Extended**

**R1(config)#IP Access-list Extended <Name>**

**R1(config)#**ip access-list extended sun

**R1(config-ext-nacl)#<Action><protocol><SIP><SWCM><DIP><DWCM>eq<PORT No./Name>**

**R1(config-ext-nacl)#**deny TCP 10.0.0.10 0.0.0.0 30.0.0.10 0.0.0.0 eq www

**R1(config-ext-nacl)#**permit TCP 10.0.0.10 0.0.0.0 30.0.0.10 0.0.0.0 eq telnet

**R1(config-ext-nacl)#**deny ICMP any any

**Exercise 13.1.** Implement all the concepts of ACL covered in Lab on Simulator.  **[10]**

# EXPERIMENT 14 – SOCKET AND SERVER SOCKET PROGRAMMING

**Objective**

* Students will able to trace the ports of a particular host.
* Students will able to Implement of echo client/server application

**Time Required :** 3 hrs

**Programming Language :** Visual C#

**Software Required :** Microsoft Visual Studios

**Hardware Required :** NIL

C# simplifies the network programming through its namespaces like System.Net and System.Net.Sockets . A Socket is an End-Point of, to and From (Bidirectional) communication link between two programs (Server Program and Client Program) running on the same network. We need two programs for communicating a socket application in C#. A Server Socket Program (Server) and a Client Socket Program (Client).

## C# socket example

[C# Server Socket Program](http://csharp.net-informations.com/communications/csharp-server-socket.htm): A C# Server Socket Program running on a computer has a socket that bound to a Port Number on the same computer and listening to the client's incoming requests.

[C# Client Socket Program](http://csharp.net-informations.com/communications/csharp-client-socket.htm): A C# Client Socket Program have to know the IP Address ( Hostname ) of the computer that the C# Server Socket Program resides and the Port Number assign for listening for client's request .

Once the connection is established between Server and Client, they can communicate (read or write) through their own sockets.

**server**

**Client**

Connection request

port

**server**

**Client**

Connection

port

port

port

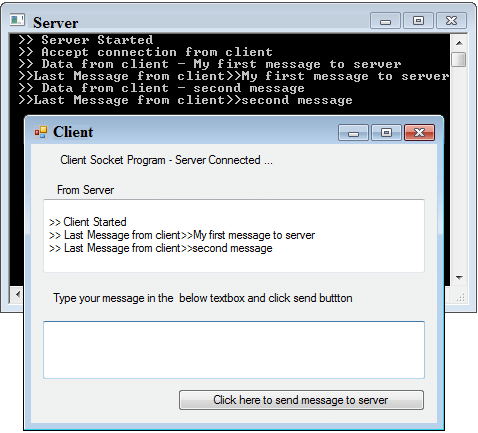
[a]: a client making a connection request to the server

[b]: session established with temporary ports used for two way communication.

listen

listen

There are two types of communication protocol uses for Socket Programming in C#, they are TCP/IP (Transmission Control Protocol/Internet protocol ) Communication and UDP/IP ( User Datagram Protocol/Internet protocol ) Communication .



The above picture shows a Server and Client communication interfaces in C#.

**[C# Server Socket Program](http://csharp.net-informations.com/communications/csharp-server-socket.htm):**

The Server Socket Program is done through a C# Console based application. Here the Server is listening for the Client's request, and when the C# Server gets a request from Client socket, the Server sends a response to the Client.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Net.Sockets;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

TcpListener serverSocket = new TcpListener(8888);

int requestCount = 0;

TcpClient clientSocket = default(TcpClient);

serverSocket.Start();

Console.WriteLine(" >> Server Started");

clientSocket = serverSocket.AcceptTcpClient();

Console.WriteLine(" >> Accept connection from client");

requestCount = 0;

while ((true))

{

try

{

requestCount = requestCount + 1;

NetworkStream networkStream = clientSocket.GetStream();

byte[] bytesFrom = new byte[10025];

networkStream.Read(bytesFrom, 0, (int)clientSocket.ReceiveBufferSize);

string dataFromClient = System.Text.Encoding.ASCII.GetString(bytesFrom);

dataFromClient = dataFromClient.Substring(0, dataFromClient.IndexOf("$"));

Console.WriteLine(" >> Data from client - " + dataFromClient);

string serverResponse = "Last Message from client" + dataFromClient;

Byte[] sendBytes = Encoding.ASCII.GetBytes(serverResponse);

networkStream.Write(sendBytes, 0, sendBytes.Length);

networkStream.Flush();

Console.WriteLine(" >> " + serverResponse);

}

catch (Exception ex)

{

Console.WriteLine(ex.ToString());

}

}

clientSocket.Close();

serverSocket.Stop();

Console.WriteLine(" >> exit");

Console.ReadLine();

}

}

}

**[C# Client Socket Program](http://csharp.net-informations.com/communications/csharp-client-socket.htm):**

The C# Client Socket Program is a windows based application . When the C# Client program execute , it will establish a connection to the C# Server program and send request to the Server , at the same time it also receive the response from C# Server .

using System;

using System.Windows.Forms;

using System.Net.Sockets;

using System.Text;

namespace WindowsApplication1

{

public partial class Form1 : Form

{

System.Net.Sockets.TcpClient clientSocket = new System.Net.Sockets.TcpClient();

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

msg("Client Started");

clientSocket.Connect("127.0.0.1", 8888);

label1.Text = "Client Socket Program - Server Connected ...";

}

private void button1\_Click(object sender, EventArgs e)

{

NetworkStream serverStream = clientSocket.GetStream();

byte[] outStream = System.Text.Encoding.ASCII.GetBytes(textBox2.Text + "$");

serverStream.Write(outStream, 0, outStream.Length);

serverStream.Flush();

byte[] inStream = new byte[10025];

serverStream.Read(inStream, 0, (int)clientSocket.ReceiveBufferSize);

string returndata = System.Text.Encoding.ASCII.GetString(inStream);

msg(returndata);

textBox2.Text = "";

textBox2.Focus();

}

public void msg(string mesg)

{

textBox1.Text = textBox1.Text + Environment.NewLine + " >> " + mesg;

}

}

}

**How to run this program?**

The C# Socket Program has two sections.

1. [C# Server Socket Program](http://csharp.net-informations.com/communications/csharp-server-socket.htm)

2. [C# Client Socket Program](http://csharp.net-informations.com/communications/csharp-client-socket.htm)

When you finish coding and build the Server and Client program , First you have to start C# Server Socket Program from DOS prompt , then you will get a message "Server Started" in your DOS screen, where the server program is running .

Next step is to start C# Client Socket Program in the same computer or other computers on the same network. When you start the client program, it will establish a connection to the Server and get a message in client screen “Client Started " , at the same time you can see a message in the Server screen "Accept connection from client" .

Now your C# Server Socket Program and C# Client Socket Program is get connected and communicated. If you want to communicate the Server and Client again, click the button in the client program, then you can see new messages in the Server and Client programs displayed.

**Exercise 14.1.** Implement the concepts of TCP socket programming in C# as discussed in this Lab. **[5]**

**Exercise 14.2.** Implement the concepts of UDP socket programming in C#.  **[5]**

**References:**

<http://csharp.net-informations.com/communications/csharp-socket-programming.htm>

|  |  |
| --- | --- |
|  | |
|  | |
|  |  |
| **“I insist you to strive. Work, Work and only work for satisfaction with patience, humbleness and serve thy nation”.** | **Muhammad Ali Jinnah** |
| **“[A scientist in his laboratory is not a mere technician: he is also a child confronting natural phenomena that impress him as though they were fairy tales”.](http://www.brainyquote.com/quotes/quotes/m/mariecurie399985.html?src=t_laboratory" \o "view quote)** | **[Marie Curie](http://www.brainyquote.com/quotes/authors/m/marie_curie.html" \o "view author)** |
| **“I have not failed. I've just found 10,000 ways that won't work”.** | **Thomas Alva Edison** |