

LABORATORY WORK BOOK

For Academic Session _____

Semester _____

TELECOMMUNICATION NETWORKS **(TC-316)**

For

TE (TC)

Name: _____

Roll Number: _____

Batch: _____

Department: _____

Year/Semester: _____



Department of Electronic Engineering
NED University of Engineering & Technology, Karachi

LABORATORY WORK BOOK
For The Course
TC-316 TELECOMMUNICATION NETWORKS

Prepared By:

Saba Ahmed (Assistant Professor)

Sundus Ali (Lecturer)

Syed Muneeb Ahmed (Lecturer)

Approved By:

Board of Studies of Department of Electronic Engineering

INTRODUCTION

Presently, is the era of very powerful personal computing devices, interconnecting with each other and even better equipped servers, connecting us across continental boundaries. Even Telecommunication Equipment is using IP backbone for providing services to its subscribers. Keeping this in mind, the lab manual for Telecommunication Networks is designed to augment the classroom teaching of the course so as to provide student the necessary practical know-how in the subject.

This Laboratory Manual will help the students learn about the design and implementation of telecommunication and data communication networks. The manual also contains some laboratory sessions on how to manage communication networks through network discovery, monitoring and management. The topics covered in this Lab Workbook include VLSM, Router and switch configuration, configuring Routing Protocols, VLANs, network discovery, packet capture and monitoring, and SNMP MIB Browsing. These practicals solidify both the theoretical and practical concepts that are very essential for the engineering students.

CONTENTS

Lab No.	DATE	Experiment	Page No.	Remarks/ Signature
1		Making Straight Through & Cross UTP Cables	1	
2		Learning IPv4 Addressing & Sub-netting (Class C Addresses)	8	
3		Learning Sub-netting (Class B & A addresses) & VLSM	12	
4		Practicing some basic Network Commands and Network Configuration Commands using command prompt and packet tracer	17	
5		Configuring static routes on Cisco routers	23	
6		Configuring RIP (Routing Information Protocol) and RIP version 2	27	
7		Configuring OSPF (Open Shortest Path First) Single Area	33	
8		Configuring EIGRP (Enhanced Interior Gateway Routing Protocol)	37	
9		Studying basic LAN switch operation	41	
10		Study and configure Access Lists	44	
11		Introduction to Network Management Tools. Learn how Networks can be managed using these tools.	47	
12		Introduction to WireShark (Network Protocol Analyzer /Packet Sniffer) and Layered Protocol	53	
13		(a) To study the concept of a Network Monitoring System (NMS). (b) To study a SNMP based NMS and observe communication between a managed element (CISCO 2950 Catalyst Switch) and NMS system	58	
14		To study the MIB(Management Info. Base) File structure written in ASN.1 focusing on RFC-1155 SMI and RFC 1212.	61	
15		To design a SNMP based managed network in Packet Tracer 5.3 using SNMPv2c and RW and Read Only communities	65	

LAB SESSION 01

OBJECT

Making the following kinds of UTP cables:

- *Straight through cable*
- *Cross cable*

THEORY

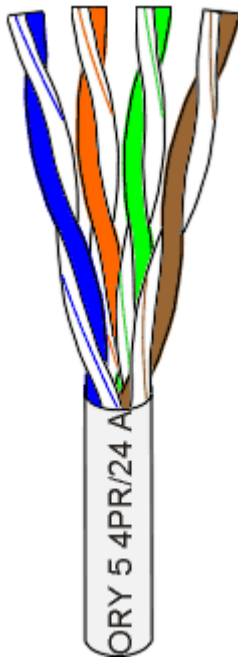


Figure 1.1:

There are several classifications of twisted pair cable. Let's skip right over them and state that we'll use Category 5 (or CAT 5) cable for all new installations. Likewise, there are several fire code classifications for the outer insulation of CAT 5 cable. We'll use CMR cable, or "riser cable," for most of the wiring we do. You should also be aware of CMP or plenum cable (a plenum is used to distribute air in a building) you may be required by local or national codes to use the more expensive plenum-jacketed cable if it runs through suspended ceilings, ducts, or other areas, if they are used to circulate air or act as an air passage from one room to another. If in doubt, use plenum. CMR cable is generally acceptable for all applications not requiring plenum cable.

CAT 5 cable is available in reel-in-box packaging. This is very handy for pulling the wire without putting twists in it. Without this kind of package or a cable reel stand, pulling wire is a two-person job. Before the advent of the reel-in-box, we used to put a reel of wire on a broom handle to pull it. One person would hold the broom handle and the other would pull broom handle to pull it. You will produce a tangled mess, if you pull the wire off the end of the reel alone.

UTP cable

Standard wire patch cables are often specified for cable segments running from a wall jack to a PC and for patch panels. They are more flexible than solid core wire. However, the rationale for using it is that the constant flexing of patch cables may wear-out solid core cable and break it. This is not a real concern in the average small network.

Most of the wiring we do simply connects computers directly to other computers or hubs. Solid core cable is quite suitable for this purpose and for many home and small business network. It is also quite acceptable for use as patch cables. You might consider a stranded wire patch cable if you have a notebook computer you are constantly moving around.

CAT 5 cable has four twisted pairs of wire for a total of eight individually insulated wires. Each pair is color coded with one wire having solid color (blue, orange, green, or brown) twisted around a second wire with a white background and a stripe of the same color. The solid color may have white stripe in some cables. Cable colors are commonly described using the background color followed by the color of the stripe; e.g; white-orange is a wire with a white background and an orange stripe.

Connectors

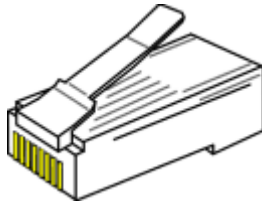


Figure 1.2: RJ-45 Connector

The straight through and cross-over patch cables are discussed in this article which is terminated with CAT 5 RJ-45 modular plugs. RJ-45 plugs are similar to those you'll see on the end of your telephone cable except they have eight as opposed to four or six contacts on the end of the plug and they are about twice as big. Make sure they are rated for CAT 5 wiring. (RJ stands for "Registered Jack"). Also, there are RJ-45 plugs designed for both solid core wire and stranded wire. Others are designed specifically for one kind of wire or the other. Be sure you buy plugs appropriate for the wire you are going to use. We normally use plugs designed to accommodate both kinds of wire.

Network cabling tools

1. Modular Plug Crimp Tool

You will need a modular crimp tool. This is very similar to the ones which have been used for many years for all kinds of telephone cable work and it works just fine for Ethernet cables. You don't need a lot of bells and whistles, just a tool which will securely crimp RJ-45 connectors. Some crimpers have cutters which can be used to cut the cable and individual wires, and possibly stripping the outer jacket.



Figure 1.3: Modular plug crimp tool

2. Universal UTP Stripping Tool (Eclipse)

It makes a much neater cut. It is highly recommending for anyone who will make a lot of cables.



Figure 1.4: Eclipse

3. Diagonal Cutters

It is easier to use diagonal cutters ("diags" or "dikes") to cut the cable off at the reel and to fine-tune the cable ends during assembly. Also, if you don't have a stripper, you can strip the cable by using a small knife to carefully slice the outer jacket longitudinally and use the diags to cut it off around the circumference.



Figure 1.5: Diagonal cutters

UTP basics

The 10BASE-T and 100BASE-TX Ethernet consist of two transmission lines. Each transmission line is a pair of twisted wires. One pair receives data signals and the other pair transmits data signals. A balanced line driver or transmitter is at one end of one of these lines and a line receiver is at the other end. A (much) simplified schematic for one of these lines and its transmitter and receiver follows:

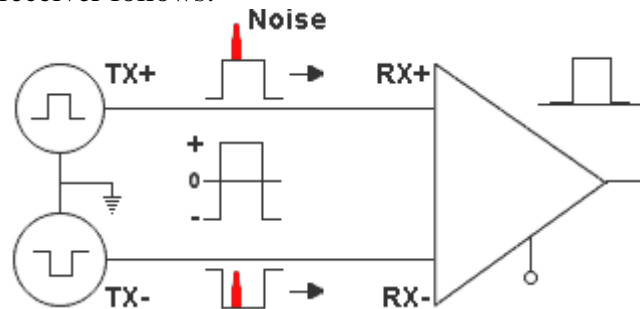


Figure 1.6: Schematic diagram of transmission line

Pulses of energy travel down the transmission line at about the speed of light (186,000 miles/second). The principal components of these pulses of energy are the potential difference between the wires and the current flowing near the surface of the wires. This energy can also be considered as residing in the magnetic field which surrounds the wires and the electric field between the wires. In other words, an electromagnetic wave which is guided by, and travels down the wires.

The main concern are the transient magnetic fields which surround the wires and the magnetic fields generated externally by the other transmission lines in the cable, other network cables, electric motors, fluorescent lights, telephone and electric lines, lightning, which may literally bury the Ethernet pulses, the conveyor of the information being sent down the line.

The twisted-pair Ethernet employs two principal means for combating noise. The first is the use of balanced transmitters and receivers. A signal pulse actually consists of two simultaneous pulses relative to ground: a negative pulse on one line and a positive pulse on the other. The receiver detects the total difference between these two pulses. Since a pulse of noise usually produces pulses of the same polarity on both lines, it is essentially canceled out at the receiver. Also, the magnetic field surrounding one wire from a signal pulse is a mirror of the one on the other wire. At a very short distance from the two wires the magnetic fields are opposite and have a tendency to cancel the effect of each other out. This reduces the line's impact on the other pairs of wires and the rest of the world.

The second and the primary means of reducing cross-talk (the term cross-talk came from the ability to overhear conversations on other lines on your phone) between the pairs in the cable, is the double helix configuration produced by twisting the wires together. This configuration produces symmetrical (differential) noise signals in each wire. Ideally, their difference as detected at the receiver, is zero. In actuality it is much reduced.

Straight through and cross over cable

Again, the wire with colored backgrounds may have white stripes and may be denoted that way in diagrams found elsewhere. For example, the green wire may be labeled Green-White. The background color is always specified first.

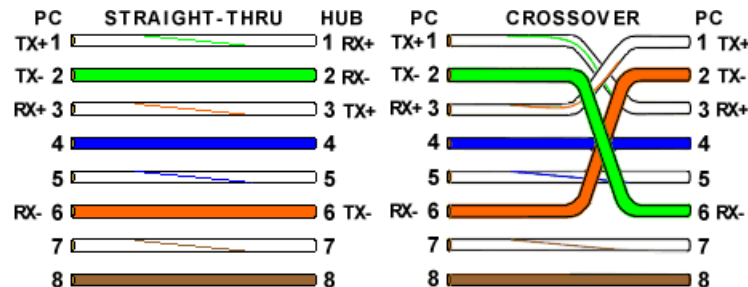


Figure 1.7: Straight through and crossover cable wire scheme

A Straight-through cable has identical ends, whereas a Crossover cable has different ends.

EIA/TIA 568A and 568B standards

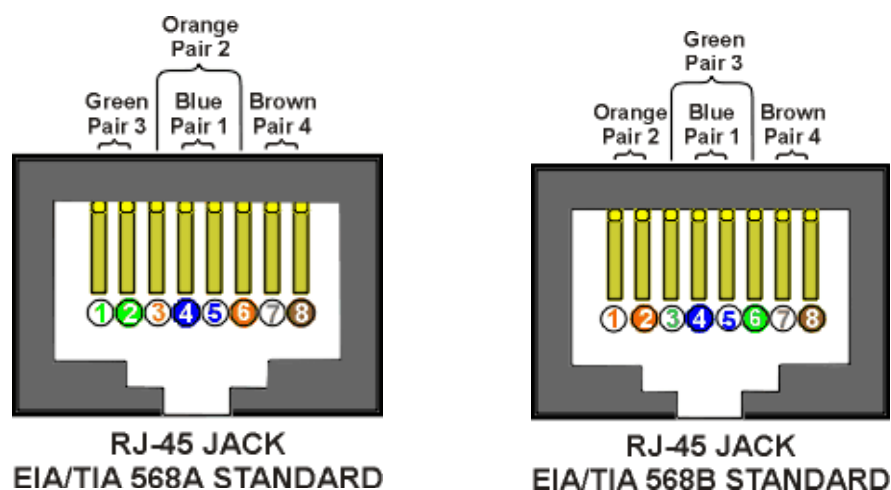


Figure 1.8: Cable connector standard ordering

It makes no functional difference which standard you use for a straight-through cable. You can start a crossover cable with either standard as long as the other end is the other standard. It makes no functional difference which end is which. Despite what you may have read elsewhere, a 568A patch cable will work in a network with 568B wiring and 568B patch cable will work in a 568A network. The electrons couldn't care less.

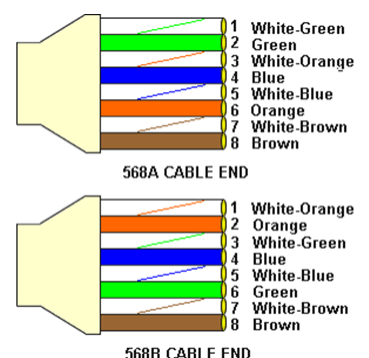


Figure 1.9: EIA/TIA 568A and 568B

PROCEDURE

To Make Cable

1. Pull the cable off the reel to the desired length and cut the total length of wire segments between a PC and a hub or between two PC's cannot exceed 100 Meters (328 feet or about the length of a football field) for 100BASE-TX and 300 Meters for 100BASE-T.
2. Strip one end of the cable with the stripper or a knife and diags. If you are using the stripper, place the cable in the groove on the blade (left) side of the stripper and align the end of the cable with the right side of the stripper. This will strip about ½" of the jacket off the cable. Turn the stripper about 1 ¼ turn and pull. If you turn it more, you will probably nick the wires. If you are using knife and diags, carefully slit the cable for about an inch or so and neatly trim around the circumference of the cable with diags to remove the jacket.
3. Inspect the wires for nicks. Cut off the end and start over if you see any. You may have to adjust the blade with the screw at the front stripper. Cable diameters and jacket thicknesses vary.
4. Spread and arrange the pairs roughly in the order of the desired cable end.
5. Untwist the pairs and arrange the wires in the order of the desired cable end. Flatten the end between your thumb and forefinger. Trim the ends of the wires so they are even with one another.

It is very important that the unstripped (untwisted) end be slightly less than ½" long. If it is longer than ½" it will be out-of-spec and susceptible to crosstalk. If it is less than ½" it will not be properly clinched when RJ-45 plug is crimped on. Flatten again. There should be little or no space between the wires.

6. Hold the RJ-45 plug with the clip facing down or away from you. Push the wire firmly into the plug. **Now, inspect before crimping and wasting the plug!** Looking through the bottom of the plug, the wire on the far-left side will have a white background. The wires should alternative light and dark from left to right. The furthest right wire is brown. The wires should all end evenly at the front of the plug. The jacket should end just about where you see it in the diagram-right on the line.

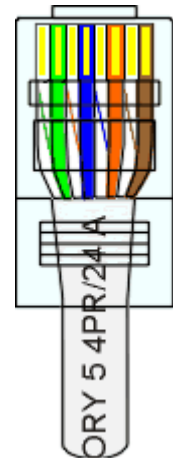


Figure 1.10:

ALL ABOUT CRIMPING

7. Hold the wire near the RJ-45 plug with the clip down and firmly push it into the left side of the front of the Crimper (it will only go in one way). Hold the wire in place and squeeze the crimper handles quite firmly. This is what will happen:

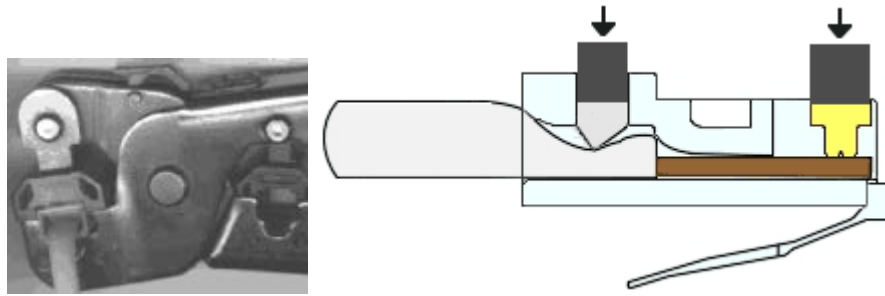


Figure 1.11: Crimping

(Crimp it once). The crimper pushes two plungers down on the RJ-45 plug. One forces, what amounts to, a cleverly designed plastic plug/wedge onto the cable jacket and very firmly clinches it. The other seats the “pins”, each with two teeth at its end, through the insulation and into the conductors of their respective wires.

8. Test the crimp... if done properly an average person will not be able to pull the plug off the cable with his or her bare hands. And that quite simply, besides lower cost, is the primary advantage of twisted-pair cables over the older thin wire, coaxial cables. In fact, the ease of installation and the modular RJ-45 plug is the main reason coaxial cable is no longer widely used for small Ethernet. But, don't pull that hard on the plug. It could stretch the cable and change its characteristics. Look at the side of the plug and see if it looks like the diagram and give it a fairly firm tug to make sure it is crimped well.
9. Prepare the other end of the cable so it has the desired end and crimp.
10. If both ends of the cable are within reach, hold them next to each other and with RJ-45 clips facing away. Look through the bottom of the plugs. If the plugs are wired correctly, and they are identical, it is a straight-through cable. If they are wired correctly and they are different, it is a crossover cable.

PRECAUTIONS

1. Try to avoid running cables parallel to power cables.
2. If you bundle a group of cables together with cable ties (zip ties), do not over-clinch them. It's okay to snug them together firmly; but don't tighten them so much that you deform the cables.
3. Keep cables away from devices which can introduce noise into them. Here's a short list: electric heaters, loud speakers, printers, TV sets, fluorescent light, copiers, welding machines, microwave ovens, telephones, fans, elevator motors, electric ovens, dryers, washing machines, and shop equipment.
4. Avoid stretching UTP cables (the force should not exceed 24 LBS).
5. Do not use a stapler to secure UTP cables. Use telephone wire hangers, which are available at most hardware stores.

EXERCISES

1. Give the reason why it is not advisable to bend UTP cables more than four times the diameter of the cable.

2. Why is it not advisable to run UTP cable outside of a building?

LAB SESSION 02

OBJECT:

Learning IPv4 Addressing & Sub-netting (Class C Addresses)

THEORY:

IP ADDRESS & SUBNET MASK

An IP (Internet Protocol) address uniquely identifies a node or host connection to an IP network. System administrators or network designers assign IP addresses to nodes. IP addresses are configured by software and are not hardware specific. An IP address is a 32 bit binary number usually represented as four fields each representing 8 bit numbers in the range 0 to 255 (sometimes called octets) separated by decimal points.

For example: 150.215.17.9

It is sometimes useful to view the values in their binary form.

150.215.17.9

10010110.11010111.00010001.00001001

An IP address consists of two parts, one identifying the network and one identifying the node. The class of the address determines which part belongs to the network address which part belongs to the node address.

A Subnet mask is a 32-bit number that masks an IP address, and divides the IP address into network address and host address. Subnet Mask is made by setting network bits to all "1"s and setting host bits to all "0"

# Bits	1	7	24		
Class A:	0	NETWORK#	HOST#		
# Bits	1	1	14	16	
Class B:	1	0	NETWORK#	HOST#	
# Bits	1	1	1	21	8
Class C:	1	1	0	NETWORK#	HOST#

An IP address has two components, the network address and host address (<network><host>)

A Subnet mask is a 32-bit number that masks an IP address, and divides the IP address into network address and host address. Subnet Mask is made by setting network bits to all "1"s and setting host bits to all "0"s.

CLASSFUL ADDRESSING

IPv4 addressing used the concept of classes. This architecture is called classful addressing. The address space is divided into five classes: A, B, C, D, and E. Each class occupies some

part of the address space.

We can find the class of an address when given the address in binary notation or dotted-decimal notation. If the address is given in binary notation, the first few bits can immediately tell us the class of the address. If the address is given in decimal-dotted notation, the first byte defines the class.

Class	Starts with	Binary range	Decimal Value range	Maximum subnets	Maximum hosts	Routing mask
A	0	00000000-01111111	0-127*	127	16,777,214	255.0.0.0
B	10	10000000-10111111	128-191	16,384	65,534	255.255.0.0
C	110	11000000-11011111	192-223	2,097,152	254	255.255.255.0
D	1110	11100000-11101111	224-239			
E	1111	11110000-11111111	240-255			

* The 0 octet is forbidden in the RFC, and 127 is reserved for loopback testing.

Network & Broadcast Addresses

- An IP address such as 176.10.0.0 that has all binary 0s in the host bit positions is reserved for the network address.
- An IP address such as 176.10.255.255 that has all binary 1s in the host bit positions is reserved for the broadcast address.

SUB-NETTING

To create a subnet address, a network administrator borrows bits from the original host portion and designates them as the subnet field.

Consider the following example:

11111111.11111111.11110000.00000000

Class B Network
16 bits for the Network
4 bits for the Subnetwork
12 bits for the Host

- ◆ 32 bits long
- ◆ Divided into four octets
- ◆ Network and subnet portions all 1's
- ◆ Host portion all 0's

Sub-netting Class C Addresses:

In a Class C address, only 8 bits are available for defining the hosts. Remember that subnet bits start at the left and go to the right, without skipping bits. This means that the only Class C subnet masks can be the following:

Binary (4th Octect)	Decimal (4th Octect)	CIDR (Classless Inter-Domain Routing) or slash notation
10000000	128	/25
11000000	192	/26
11100000	224	/27
11110000	240	/28
11111000	248	/29
11111100	252	/30

Now determine the following:

How many subnets? 2^x = number of subnets. x is the number of masked bits, or the 1s. For example, in 11000000, the number of ones gives us 2^2 subnets. In this example, there are 4 subnets.

How many hosts per subnet? $2^y - 2$ = number of hosts per subnet. y is the number of unmasked bits, or the 0s. For example, in 11000000, the number of zeros gives us $2^6 - 2$ hosts. In this example, there are 62 hosts per subnet. You need to subtract two for the subnet address and the broadcast address, which are not valid hosts.

What are the valid subnets? $256 - \text{Subnet mask} = \text{block size, or increment number}$. An example would be $256 - 192 = 64$. The block size of a 192 mask is always 64. Start counting at zero in blocks of 64 until you reach the subnet mask value and these are your subnets. 0, 64, 128, 192.

What's the broadcast address for each subnet? Since we counted our subnets in the last section as 0, 64, 128, and 192, the broadcast address is always the number right before the next subnet. For example, the 0 subnet has a broadcast address of 63 because the next subnet is 64. The 64 subnet has a broadcast address of 127 because the next subnet is 128, etc. The broadcast of the last subnet is always 255 for Class C.

What are the valid hosts? Valid hosts are the numbers between the subnets, omitting all the 0s and all 1s. For example, if 64 is the subnet number and 127 is the broadcast address, then 65–126 is the valid host range—it's *always* the numbers between the subnet address and the broadcast address.

EXERCISE:

1. Find the class of each address.

- 00000001 00001011 00001011 11101111
- 11000001 10000011 00011011 11111111
- 14.23.120.8
- 252.5.15.111

- a. 192.168.10.0 (/26)
- b. 192.168.10.0 (/27)

This image shows a single sheet of white paper with horizontal blue or grey ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

LAB SESSION 03

OBJECT:

Learning Sub-netting (Class B & A addresses) & VLSM

THEORY

Sub-netting Class B Addresses

Binary (3rd and 4th Octet)	Decimal (All Octets)	CIDR (Classless Inter-Domain Routing) or slash notation
10000000 00000000	255.255.128.0	/17
11000000 00000000	255.255.192.0	/18
11100000 00000000	255.255.224.0	/19
11110000 00000000	255.255.240.0	/20
11111000 00000000	255.255.248.0	/21
11111100 00000000	255.255.252.0	/22
11111110 00000000	255.255.254.0	/23
11111111 00000000	255.255.255.0	/24
11111111 10000000	255.255.255.128	/25
11111111 11000000	255.255.255.192	/26
11111111 11100000	255.255.255.224	/27
11111111 11110000	255.255.255.240	/28
11111111 11111000	255.255.255.248	/29
11111111 11111100	255.255.255.252	/30

Then determine all the parameters discussed in Lab 02 in Sub-netting Class C Address section.

Sub-netting Class A Addresses

Note that the Class A addresses can be sub-netted in the same way as for Class C & B. However, in that case we have sub-netting possible in 3 octets as opposed to 1 or 2 subnets as in Class C or B respectively.

VARIABLE LENGTH SUBNET MASKING (VLSM)

Variable Length Subnet Masking (VLSM) is a way of further sub-netting a subnet. Using Variable Length Subnet Masking (VLSM) we can allocate IP addresses to the subnets by the exact need (*in the power of 2*).

Variable Length Subnet Masking (VLSM) allows us to use more than one subnet mask within the same network address space.

If we recollect from the previous lessons, we can divide a network only into subnets with equal number of IP addresses. Variable Length Subnet Masking (VLSM) allows creating subnets from a single network with unequal number of IP addresses.

Example: We want to divide 192.168.10.0, which is a Class C network, into four networks, each with unequal number of IP address requirements as shown below.

Subnet A : 126 IP Addresses

Subnet B : 62 IP Addresses

Subnet C : 30 IP Addresses

Subnet D : 30 IP Addresses

Original Network (Network to be subnetted) – 192.168.10.0/24

(VLSM) - First Division

Divide the two networks equally with 128 [IPv4 addresses](#) (126 usable [IPv4 addresses](#)) in each network using 255.255.255.128 [subnet mask](#) (192.168.10.0/25).

We will get two subnets each with 128 [IPv4 addresses](#) (126 usable [IPv4 addresses](#)).

1) 192.168.10.0/25, which can be represented in binaries as below.

11000000.10101000.00001010.00000000

11111111.11111111.11111111.10000000

2) 192.168.10.128/25, which can be represented in binaries as below.

11000000.10101000.00001010.10000000

11111111.11111111.11111111.10000000

(VLSM)- Second Division

Divide second subnet (192.168.10.128/25) we got from the first division again into two Networks, each with 64 IP Addresses (62 usable [IPv4 addresses](#)) using 255.255.255.192 [subnet mask](#).

We will get two subnets each with 64 [IPv4 addresses](#) (62 usable [IPv4 addresses](#)).

1) 192.168.10.128/26, which can be represented in binaries as below

11000000.10101000.00001010.10000000

11111111.11111111.11111111.11000000

2) 192.168.10.192/26

11000000.10101000.00001010.11000000

11111111.11111111.11111111.11000000

(VLSM) - Third Division

Divide 192.168.10.192/26 Network again into two Networks, each with 32 IPv4 addresses (30 usable IPv4 addresses) using 255.255.255.224 subnet mask

We will get two subnets each with 32 IPv4 addresses (30 usable IPv4 addresses).

1) 192.168.10.192/27, which can be represented in binaries as below.

11000000.10101000.00001010.11000000

11111111.11111111.11111111.11100000

2) 192.168.10.224/27, which can be represented in binaries as below.

11000000.10101000.00001010.11100000

11111111.11111111.11111111.11100000

Now we have split the 192.168.10.0/24 network into four subnets using Variable Length Subnet Masking (VLSM), with unequal number of IPv4 addresses as shown below. Also note that when you divide a network using Variable Length Subnet Masking (VLSM), the subnet masks are also different.

1)	192.168.10.0	255.255.255.128	(126	(128-2) IP v4 Addresses)
2)	192.168.10.128	255.255.255.192	(62	(64-2) IP v4 Addresses)
3)	192.168.10.192	255.255.255.224	(30	(32-2) IP v4 Addresses)
4)	192.168.10.224	255.255.255.224	(30	(32-2) IP v4 Addresses)

EXERCISES

1. Subnets the following addresses and verify your results using any online IPv4 Addressing & Sub-netting Calculator and attach their screen shots.

a. 172.16.0.0 (/19)

b. 10.0.0.0 (/10)

[illegible]

-
- This image shows a full page of white paper with horizontal blue ruling lines. The lines are evenly spaced and run across the width of the page, providing a template for handwriting practice or general writing. There are no margins, text, or other markings on the page.

LAB SESSION 04

OBJECT:

Practicing some basic Network Commands and Network Configuration Commands using command prompt and packet tracer

THEORY:

ipconfig:

ipconfig (internet protocol configuration) in Microsoft Windows is a console application that displays all current TCP/IP network configuration values and can modify Dynamic Host Configuration Protocol DHCP and Domain Name System DNS settings.

ping:

Verifies IP-level connectivity to another TCP/IP computer by sending Internet Control Message Protocol (ICMP) Echo Request messages. The receipt of corresponding Echo Reply messages are displayed, along with round-trip times. Ping is the primary TCP/IP command used to troubleshoot connectivity, reachability, and name resolution. Used without parameters, ping displays help.

tracert:

tracert is a command-line tool included with Windows and other operating systems. Along with the ping command, it's an important tool for understanding Internet connection problems, including packet loss and high latency.

If you're having trouble connecting to a website, tracert can tell you where the problem is. It can also help visualize the path traffic takes between your computer and a web server.

nslookup:

nslookup is a network administration command-line tool available for many computer operating systems for querying the Domain Name System (DNS) to obtain domain name or IP address mapping or for any other specific DNS record.

Modes and CLI commands for Routers

Cisco routers support different modes of operation. When you access a router, it will typically be in the **"user mode"**. User mode gives a user access to simple "show commands." From user mode the next step is **"Privileged mode."** In the "Privileged mode" a user can have full access to all the databases maintained by the router. **Configuration Mode** in which we can configure the router Cisco routers use many other modes, but let us keep it simple for now.

User mode is identified by prompt ending with ">" to switch to privileged mode type "enable" the prompt should end with # e.g. Router#

Configuring the Router

You will be able to learn the basic commands for configuring a router.

sh running-config - details the running configuration file (RAM)

sh startup-config - displays the configuration stored in NVRAM

setup - Will start the automatic setup; the same as when you first boot the router

config t - use to execute configuration commands from the terminal

config mem - executes configuration commands stored in NVRAM; copies startup-config to running-config

config net - used to retrieve configuration info from a TFTP server

copy running-config startup-config - copies saved config in running config (RAM) to NVRAM or "write memory" for IOS under ver.11

copy startup-config running-config - copies from non-volatile (NVRAM) to current running config (RAM)

boot system flash <put file filename here> - tells router which IOS file in flash to boot from

boot system tftp - tells router which IOS file on the tftp server to boot from

boot system rom - tell router to boot from ROM at next boot

copy flash tftp - Copies flash to tftp server

copy tftp flash - Restores flash from tftp server

copy run tftp - Copies the current running-config to tftp server

copy tftp run - Restores the running-config from tftp server

General Commands

Here is a list of the general commands. These are the basic level commands and most commonly used

no shutdown - (enables the interface)

reload - restarts the router

sh ver - Cisco IOS version, uptime of router, how the router started, where system was loaded from, the interfaces the POST found, and the configuration register

sh clock - shows date and time on router

sh history - shows the history of your commands

sh debug - shows all debugging that is currently enabled

no debug all - turns off all debugging

sh users - shows users connected to router

sh protocols - shows which protocols are configured

banner motd # Your customized message here # - Set/change banner

hostname <give router name> - use to configure the hostname of the router

clear counters - clear interface counters

Privileged Mode commands of a router

Learn how to work in the privileged mode of a router.

enable - get to privileged mode

disable - get to user mode

enable password <give password here> - sets privileged mode password

enable secret <give password here> - sets encrypted privileged mode password

Setting Passwords on router

Here you will be able to learn how to set the password on a router.

enable secret <give password here> - set encrypted password for privileged access

enable password <give password here> - set password for privileged access (used when

there is no enable secret and when using older software)

Setting the password for console access:

```
(config)#line console 0
```

```
(config-line)#login
```

```
(config-line)#password <put password here>
```

Set password for virtual terminal (telnet) access (password must be set to access router through telnet):

```
(config)#line vty 0 4
```

```
(config-line)#login
```

```
(config-line)#password <put password here>
```

Set password for auxiliary (modem) access:

```
(config)#line aux 0
```

```
(config-line)#login
```

```
(config-line)#password <put password here>
```

Router Processes & Statistics

By these command you can see the statistics and different processes of the router.

sh processes - shows active processes running on router

sh process cpu - shows cpu statistics

sh mem - shows memory statistics

sh flash - describes the flash memory and displays the size of files and the amount of free flash memory

sh buffers - displays statistics for router buffer pools; shows the size of the Small, Middle, Big, Very Big, Large and Huge Buffers

sh stacks - shows reason for last reboot, monitors the stack use of processes and interrupts routines

IP Commands

Here is a list of the IP Commands

Configure IP on an interface:

```
int serial 0
```

```
ip address 157.89.1.3 255.255.0.0
```

```
int eth 0
```

```
ip address 2008.1.1.4 255.255.255.0
```

Other IP Commands:

sh ip route - view ip routing table

ip route <remote_network> <mask> <default_gateway> [administrative_distance] - configure a static IP route

ip route 0.0.0.0 0.0.0.0 <put gateway of the last resort here> - sets default gateway

ip classless - use with static routing to allow packets destined for unrecognized subnets to use the best possible route

sh arp - view arp cache; shows MAC address of connected routers

ip address 2.2.2.2 255.255.255.0 secondary - configure a 2nd ip address on an interface

sh ip protocol

CDP Commands (Cisco Discovery Protocol uses layer 2 multicast over a SNAP-capable link to send data):

sh cdp neighbor - shows directly connected neighbors
sh cdp int - shows which interfaces are running CDP
sh cdp int eth 0/0 - show CDP info for specific interface
sh cdp entry <cdp neighbor here> - shows CDP neighbor detail
cdp timer 120 - change how often CDP info is sent (default cdp timer is 60)
cp holdtime 240 - how long to wait before removing a CDP neighbor (default CDP holdtime is 180)
sh cdp run - shows if CDP turned on
no cdp run - turns off CDP for entire router (global config)
no cdp enable - turns off CDP on specific interface

IPX Commands

Enable IPX on router:

ipx routing

Configure IPX + IPX-RIP on an int:

int ser 0

ipx network 4A

Other Commands:

sh ipx route - shows IPX routing table
sh ipx int e0 - shows ipx address on int
sh ipx servers - shows SAP table
sh ipx traffic - view traffic statistics
debug ipx routing activity - debugs IPS RIP packets
debug ipx sap - debugs SAP packets

Routing Protocols:

RIP, IGPR and OSPF are the routing protocols and here is a list of the commands for the working on the routing protocols.

Configure RIP:

router rip

network 157.89.0.0

network 208.1.1.0

Other RIP Commands:

debug ip rip - view RIP debugging info

Configure IGRP:

router IGRP 200

network 157.89.0.0

network 208.1.1.0

Other IGRP Commands:

debug ip igrp events - view IGRP debugging info

debug ip igrp transactions - view IGRP debugging info

Access Lists

Here is a list of the Access list command of a router.

sh ip int ser 0 - use to view which IP access lists are applies to which int

sh ipx int ser 0 - use to view which IPX access lists are applies to which int


```
ipx access-group 900 out
```

3. Using nslookup find ip address of common social media sites.

LAB SESSION 05

OBJECT

Configuring static routes on Cisco routers

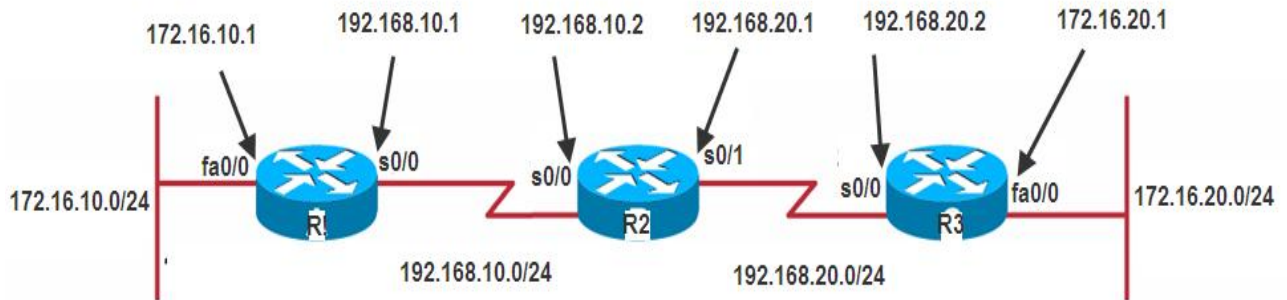


Figure 5.1: Scenario for static routes

THEORY

Routed & Routing Protocols

- A **Routed Protocol** is a protocol by which data can be routed. Routed protocols are IP, AppleTalk, and IPX. In this kind of protocols we require an addressing scheme and sub netting. Addressing scheme will be used to determine the network to which a host belongs and to identifying that host on that particular network. All hosts on an internetwork use the services of a routed protocol.
- A **Routing Protocol** is different and is only used between routers. It makes possible for routers to build and maintain routing tables. There are three classes of routing protocols-
 - 1) Distance Vector,
 - 2) Link State,
 - 3) Hybrid

Static & Dynamic Routing

The simplest method to route packets on a network is static routes. Although dynamic routing protocols are flexible and adjust to network changes, they do have associated network traffic which competes for network bandwidth with the user data traffic.

Configuring Static Routes

Static routes specify a fixed route for a certain destination network. They need to be configured on any router that needs to reach a network that it is not directly connected to.

The IOS command used to configure static routes is `ip route`. The syntax is:

```
ip route destination-address subnet-mask {ip-address | outgoing-interface}  
[distance] [tag tag] [permanent]
```

where:

- *destination-address* is the destination address prefix for the network that we would like the router to reach
- *subnet-mask* is the subnet mask to be used on the address prefix to match for destination addresses. Multiple networks may be combined such that the destination-address and subnet-mask combination matches all hosts on those networks.
- *ip-address* specifies what ip address to forward a packet to if an IP packet arrives with a destination address that matches the destination-address subnet-mask pair specified in this command.
- Alternatively *outgoing-interface* specifies which interface the packet should be sent out of. Adding a static route to an Ethernet or other broadcast interface (for example, `ip route 0.0.0.0 0.0.0.0 Ethernet 1/2`) will cause the route to be inserted into the routing table only when the interface is up. This configuration is not generally recommended. When the next hop of a static route points to an interface, the router considers each of the hosts within the range of the route to be directly connected through that interface, and therefore it will send ARP requests to any destination addresses that route through the static route.
- *distance* is the optional administrative distance value for the route. If unspecified the default value is 1.
- *tag* value can be used as a "match" value for controlling redistribution via route maps.
- *permanenet* specifies that the route will not be removed even if the interface shuts down.

DTE/DCE

DCE and DTE are the interfaces. The DCE-DTE connection between routers is referred to as a null serial cable DCE(data communication equipment) and DTE (Data terminal equipment). DCE is located at the service provider end while the DTE is attached device.

The services that are given to the DTE is often accessed via modems or channel service unit/data service unit(CSU/DSU). DCE provides clocking and DTE receives the clock

PROCEDURE

1. Connect the network as shown in the network diagram.
2. Configure appropriate ip addresses and clock rates(if needed) on the router interfaces as specified in the network diagram.
3. For R1, enter the following static routes

```
ip route 172.16.20.0 255.255.255.0 192.168.10.2  
ip route 192.168.20.0 255.255.255.0 192.168.10.2
```

- ```
ip route 172.16.10.0 255.255.255.0 192.168.10.1
ip route 172.16.20.0 255.255.255.0 192.168.20.2
```

- ```
ip route 172.16.10.0 255.255.255.0 192.168.20.1
ip route 192.168.10.0 255.255.255.0 192.168.20.1
```

- ```
router# sh ip route
```

**1. Run the command show IP route and write its output.**

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and extend across the width of the page. There are no margins, text, or other markings on the paper.

- [illegible]

- \_\_\_\_\_
- \_\_\_\_\_
- \_\_\_\_\_
- 3. Create a loop back interface on R3 and assign an IP address 10.1.0.1 /16 to it. Now add static routes to each of the other routers to reach this interface. Verify your work by pinging the newly created interface from routers R1 and R2 respectively.**

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

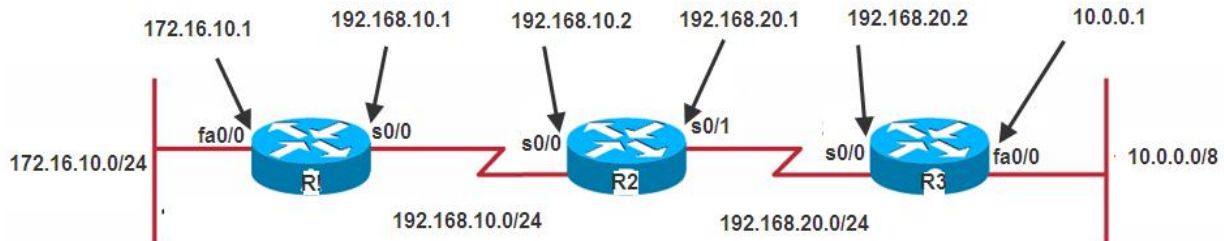
\_\_\_\_\_

\_\_\_\_\_

## **LAB SESSION 06**

### **OBJECT:**

#### ***Configuring RIP (Routing Information Protocol) and RIP version 2***



**Figure 6.1:** Scenario for RIP

### **THEORY**

#### **Distance Vector Routing Protocols**

- Broadcast their entire routing table to each neighbor router at predetermined intervals
- The actual interval depends on the distance-vector routing protocol in use
- Varies between 30 and 90 seconds
- Sometimes referred to as *routing by rumor*
- Suffer from slow time to *convergence*
- *Convergence* is a state where all routers on the internetwork share a common view of the internetwork routes

#### **Routing Information Protocol (RIP)**

A distance-vector protocol, RIP was designed to work with small to medium-sized networks. RIP is an Interior Gateway Protocol (IGP), meaning it is used within an autonomous system. An autonomous system is a collection of networks under a single administration, sharing a common routing strategy.

RIP is easy to implement, compared to newer IGPs, and has been implemented in networks around the world. Advantage of using RIP, especially in small networks, is that there is very little overhead, in terms of bandwidth used and configuration and management time.

#### **RIP Timers**

RIP uses timers both to regulate its performance and to help prevent routing loops. All routers that use RIP send an update message to all of their neighbors approximately every 30 seconds; this process is termed *advertising*. The Cisco implementation sends updates every 30 seconds minus up to 15 percent, or 4.5 seconds.

If a neighbor has not responded in 180 seconds, it is assumed that the neighboring router is

unavailable or the network connecting it to the router has become unusable. When the neighbor has not responded for 180 seconds, the route is marked invalid; 180 seconds is long enough that a route won't be invalidated by a single missed update message. The neighbor is shown to be unreachable by sending a normal update message with a metric of "infinity;" in the case of RIP, this number is 16. If an advertisement is received from a neighbor with a metric of infinity, then the route is placed into hold-down state, advertised with a distance of 16, and kept in the routing table. No updates from other neighbors for the same route are accepted while the route is in hold-down state. If other neighbors are still advertising the same route when the hold-down timer expires, then their updates will then be accepted. The route will be advertised with infinity metric for a period of time after the hold-down state if no alternate paths are found.

The actual timers used to accomplish the above tasks are a *routing-update timer*, a *route-invalid timer*, a *route-hold-down timer*, and a *route-flush timer*. The RIP routing-update timer is generally set to 30 seconds, ensuring that each router will send a complete copy of its routing table to all neighbors every 30 seconds. The route-invalid timer determines how much time must expire without a router having heard about a particular route before that route is considered invalid. When a route is marked invalid or put in hold-down state, neighbors are notified of this fact. This notification must occur prior to expiration of the route-flush timer. When the route flush-timer expires, the route is removed from the routing table. Typical initial values for these timers are 180 seconds for the route-invalid and route-holddown timers and 240 seconds for the route-flush timer. The values for each of these timers can be adjusted with the `timers basic` router configuration command.

### Several Stability Features

To adjust for rapid network-topology changes, RIP specifies numerous stability features that are common to many routing protocols. RIP implements split horizon with poison-reverse and hold-down mechanisms to prevent incorrect routing information from being propagated. Split horizon prevents incorrect messages from being propagated by not advertising routes over an interface that the router is using to reach the route. Implementing split horizon helps avoid routing loops. Poison reverse operates by advertising routes that are unreachable with a metric of infinity back to the original source of the route. Hold-down is a method of marking routes invalid (expired). As discussed above, no updates from other neighbors for the same route are accepted while the route is in hold-down state.

Triggered updates are also an included convergence and stability feature. Updates are triggered whenever a metric for a route changes. Triggered updates may also contain only information regarding routes that have changed, unlike scheduled updates.

### RIP version 2

RIPv2 is almost the same as the RIP version 1. RIPv2 also sends its complete routing table to its active interfaces at periodic time intervals. The timers, loop avoidance schemes and administrative distance are the same as Rip version 1. But RIPv2 is considered classless routing protocol because it also sends subnet information's with each router. It also allows authentication using MD5 encryption scheme. And it also supports dis-contiguous networks. Configuring RIP version 2 on a router is very simple; it just requires one additional command



## PROCEDURE

### Configuring RIP

1. Cable up the network as shown in the diagram.
2. Assign the IP address as shown in the diagram to the appropriate interfaces. For the serial links, has been used to indicate a DCE port.
3. Issue RIP routing commands on all the routers starting from the global config mode.
4. On R1:  
    router rip  
    network 172.16.10.0  
    network 192.168.10.0  
    On R2  
    router rip  
    network 192.168.10.0  
    network 192.168.20.0  
    On R 3  
    router rip  
    network 10.0.0.0  
    network 192.168.20.0
5. To verify the working of RIP ping one host, say H2, on LAN connected to R3 from the host, say H1, on LAN connected to R1. Also run some other debugging command to explore more.

### Configuring RIP version 2

1. Issue the following commands on R1.  
    Router rip version 2  
    network 172.16.10.0  
    network 192.168.10.0
2. Repeat the same for R2 and R3.
3. Verify and debug, as you did earlier for RIP.

## EXERCISES

1. **Configure RIP on all three routers, note down routing table of router R1, and run command Debug ip rip to note the address on which updates are sent.**
- 
-

---

---

---

---

---

---

---

---

---

---

---

---

**2. Write commands to modify the default update and hold-down timers.**

---

---

---

---

---

---

---

---

---

---

---

---

**3. Repeat exercise #1 for RIPv2 and note down the multicast address on which RIPv2 forwards the updates.**

---

---

---

---

---

---

---

---

---

---

---

---

---

---

**4. Write down the source IP address for the ping packets when you ping H1 from R1.**

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

**5. While working on R1, how could you check if H1 can reach the loopback interface?  
In other words, how can you verify if a ping from H1 to loopback of R1 is successful?**

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

## **LAB SESSION 07**

### **OBJECT:**

#### ***Configuring OSPF (Open Shortest Path First) Single Area***

### **THEORY**

Open Shortest Path First (OSPF) was developed by the Internet Engineering Task Force (IETF) as a replacement for the problematic RIP and is now the IETF-recommended Interior Gateway Protocol (IGP). OSPF is a link state protocol that, as the name implies, uses Dijkstra's Shortest Path First (SPF) algorithm. It is an open standards protocol—that is, it isn't proprietary to any vendor or organization. Link-state routing protocols perform the following functions:

- Respond quickly to network changes
- Send triggered updates only when a network change has occurred
- Send periodic updates known as *link-state refreshes*
- Use a *hello mechanism* to determine the reachability of neighbors
- Each router keeps track of the state or condition of its directly connected neighbors by multicasting hello packets
- Each router also keeps track of all the routers in its network or area of the network by using *link-state advertisements (LSAs)*.

Like all link state protocols, OSPF's major advantages over distance vector protocols are fast convergence, support for much larger internetworks, and less susceptibility to bad routing information. Other features of OSPF are:

- The use of areas, which reduces the protocol's impact on CPU and memory, contains the flow of routing protocol traffic, and makes possible the construction of hierarchical internetwork topologies
- Fully classless behavior, eliminating such class-full problems as dis-contiguous subnets. Support of classless route table lookups, VLSM, and super-netting for efficient address management
- A dimensionless, arbitrary metric
- Equal-cost load balancing for more efficient use of multiple paths
- Support of authentication for more secure routing
- The use of route tagging for the tracking of external routes

## **Characteristics of OSPF**

| <b>Characteristic</b>                              | <b>OSPF</b>               |
|----------------------------------------------------|---------------------------|
| VLSM support                                       | Yes                       |
| Manual summarization                               | Yes                       |
| Type of protocol                                   | Link state                |
| Classless support                                  | Yes                       |
| Auto-summarization                                 | No                        |
| Dis-contiguous support                             | Yes                       |
| Route propagation                                  | Multicast on change       |
| Hop count limit                                    | None                      |
| Convergence                                        | Fast                      |
| Peer authentication                                | Yes                       |
| Hierarchical network<br>Updates/ Route computation | Event triggered/ Dijkstra |

## **DR and BDR**

### ***DR (Designated Routers)***

DR has the following duties:

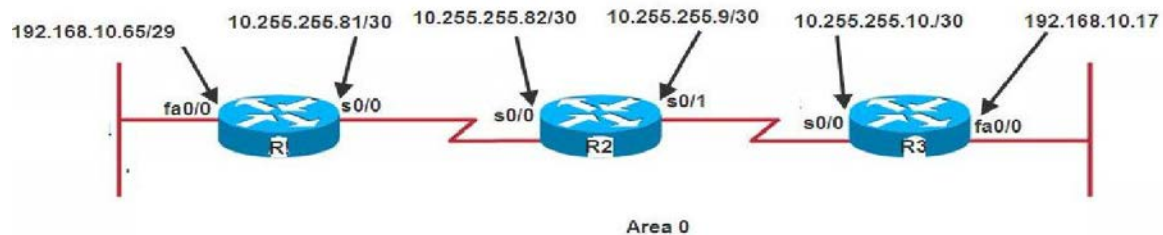
- To represent the multi-access network and its attached routers to the rest of the internetwork
- To manage the flooding process on the multi-access network.
- The concept behind the DR is that the network itself is considered a "pseudo node," or a virtual router. Each router on the network forms an adjacency with the DR which represents the pseudo-node. Only the DR will send LSAs to the rest of the internetwork.

Note: router might be a DR on one of its attached multi-access networks, and it might not be the DR on another of its attached multi-access networks. In other words, the DR is a property of a router's interface, not the entire router.

## BDR(Backup Designated Router):

A *Backup Designated Router (BDR)* is a hot standby for the DR on multi-access links. The BDR receives all routing updates from OSPF adjacent routers but doesn't flood LSA updates.

*Note: if the router interface priority value is set to zero then that router won't participate in the DR or BDR elections on that interface.*



**Fig 7.1:** Scenario for OSPF implementation

After assigning ip addresses to interfaces of the routers the following IP Routing commands of OSPF on each other will be given as below.

Router A:

```
Router_A#config t
Router_A(config)#router ospf 1
Router_A(config-router)#network 192.168.10.64 0.0.0.7 area 0
Router_A(config-router)#network 10.255.255.80 0.0.0.3 area 0
```

*The Router\_A is using a /29 or 255.255.255.248 mask on the fa0/0 interface. This is a block size of 8, which is a wildcard of 7. The s0/0 interface is a mask of 255.255.255.252 block size of 4, with a wildcard of 3. Similarly the other subnet ,mask, and wildcard can be determined by looking at the IP address of an interface.*

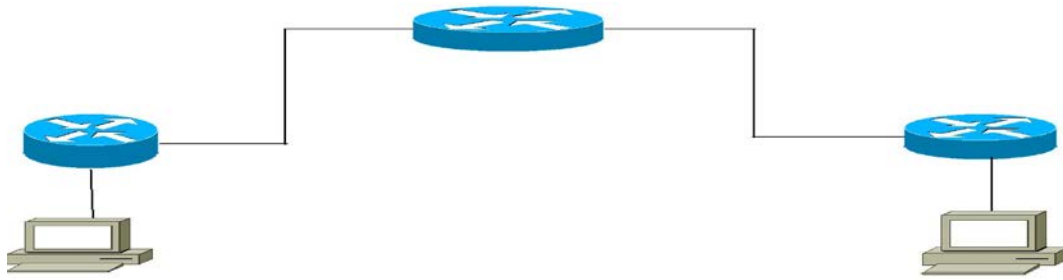
Router B:

```
Router_B#config t
Router_B(config)#router ospf 1
Router_B(config-router)#network 10.255.255.80 0.0.0.3 area 0
Router_B(config-router)#network 10.255.255.8 0.0.0.3 area 0
```

Router C:

```
Router_C#config t
Router_C(config)#router ospf 1
Router_C(config-router)#network 192.168.10.16 0.0.0.7 area 0
Router_C(config-router)#network 10.255.255.8 0.0.0.3 area 0
```

## EXERCISES



**Fig 7.2:** Scenario for exercise problems

**Simulate the network shown above on packet tracer. Assign appropriate IP addresses on the interfaces and configure OSPF on the routers. Write down the configuration commands entered on all three routers for configuration of OSPF.**

**1. Router 1:**

---

---

---

---

---

---

---

---

**2. Router 2**

---

---

---

---

---

---

---

---

**3. Router 3**

---

---

---

---

---

---

---

---



## **LAB SESSION 08**

### **OBJECT**

#### ***Configuring EIGRP (Enhanced Interior Gateway Routing Protocol)***

### **THEORY**

EIGRP is a proprietary Cisco protocol that runs on Cisco routers. It is important to understand EIGRP because it is probably one of the two most popular routing protocols in use today. Like IGRP, EIGRP uses the concept of an autonomous system to describe a set of contiguous routers that run the same routing protocol and share routing information. But unlike IGRP, EIGRP includes the subnet mask in its route updates.

#### **Why prefer EIGRP?**

EIGRP is sometimes referred to as a hybrid routing protocol because it has characteristics of both distance vector and link state protocols. For example, EIGRP doesn't send link-state packets as OSPF does; instead it sends traditional distance vector updates containing information about networks plus the cost of reaching them from the perspective of the advertising router. And EIGRP has link state characteristics as well – it synchronizes routing tables between neighbors at startup and then sends specific updates only when topology changes occur. This makes EIGRP suitable for very large networks. EIGRP has a maximum hop count of 255 (the default is set to 100).

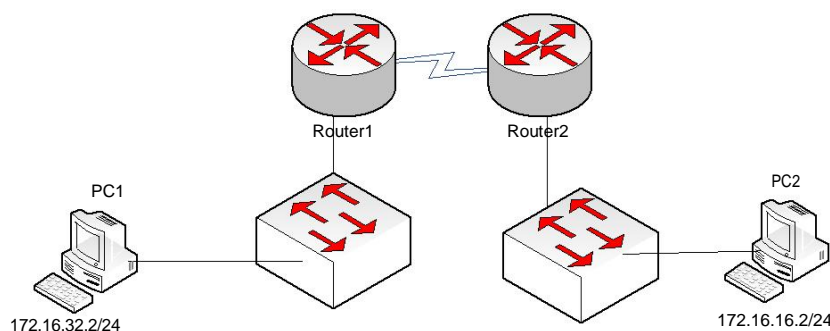
#### **EIGRP metric calculation:**

EIGRP unlike many other protocols that use a single factor to compare routes and select the best possible path, EIGRP can use a combination of four:

- 1) Bandwidth
- 2) Delay
- 3) Load
- 4) Reliability

#### **Configuring EIGRP**

Lets view the topology



**Fig 8.1:** Scenario for EIGRP implementation

Following are the IP addresses assigned to the interfaces

```
Router2#sh ip int brief
```

| Interface       | IP-Address  | OK? | Method | Status                | Protocol |
|-----------------|-------------|-----|--------|-----------------------|----------|
| FastEthernet0/0 | unassigned  | YES | Manual | up                    | down     |
| FastEthernet1/0 | 172.16.32.1 | YES | Manual | up                    | up       |
| Serial2/0       | 172.16.64.2 | YES | Manual | up                    | up       |
| Serial3/0       | unassigned  | YES | manual | administratively down | down     |
| FastEthernet4/0 | unassigned  | YES | manual | administratively down | down     |
| FastEthernet5/0 | unassigned  | YES | manual | administratively down | down     |
| Modem6/0        | unassigned  | YES | manual | down                  | down     |
| Modem7/0        | unassigned  | YES | manual | down                  | down     |
| Modem8/0        | unassigned  | YES | manual | down                  | down     |

```
Router1#sh ip int brief
```

| Interface       | IP-Address  | OK? | Method | Status                | Protocol |
|-----------------|-------------|-----|--------|-----------------------|----------|
| FastEthernet0/0 | 172.16.32.1 | YES | Manual | up                    | up       |
| FastEthernet1/0 | unassigned  | YES | Manual | administratively down | down     |
| Serial2/0       | 172.16.64.1 | YES | Manual | up                    | up       |
| Serial3/0       | unassigned  | YES | manual | administratively down | down     |
| FastEthernet4/0 | unassigned  | YES | manual | administratively down | down     |
| FastEthernet5/0 | unassigned  | YES | manual | administratively down | down     |

To start EIGRP process on both routers the following configurations will be done.

```
Router1(config)#router eigrp 1
Router1(config-router)#network 172.16.64.0 0.0.0.255
Router1(config-router)#network 172.16.32.0 0.0.0.255
Router1(config-router)#exit
```

```
Router2(config)#router eigrp 1
Router2(config-router)#network 172.16.64.0 0.0.0.255
Router2(config-router)#network 172.16.64.0 0.0.0.255
Router2(config-router)#network 172.16.16.0 0.0.0.255
Router2(config-router)#exit
```

Now verifying the routing tables.

```
Router2#sh 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
 E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
 i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
area
```

\* - candidate default, U - per-user static route, o -  
ODR P - periodic downloaded static route

Gateway of last resort is not set

```
172.16.0.0/24 is subnetted, 3 subnets
C 172.16.16.0 is directly connected, FastEthernet0/0
D 172.16.32.0 [90/20514560] via 172.16.64.1, 00:01:36, Serial2/0
C 172.16.64.0 is directly connected, Serial2/0
```

Router1#sh 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  
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP  
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter  
area  
\* - candidate default, U - per-user static route, o -  
ODR P - periodic downloaded static route

Gateway of last resort is not set

```
172.16.0.0/24 is subnetted, 3 subnets
D 172.16.16.0 [90/20514560] via 172.16.64.2, 00:01:53, Serial2/0 C
172.16.32.0 is directly connected, FastEthernet0/0
C 172.16.64.0 is directly connected, Serial2/0
```

Now we will check end to end connectivity from PCs.

PC2>ping 172.16.32.2

Pinging 172.16.32.2 with 32 bytes of data:

```
Reply from 172.16.32.2: bytes=32 time=156ms TTL=126
Reply from 172.16.32.2: bytes=32 time=125ms TTL=126
Reply from 172.16.32.2: bytes=32 time=127ms TTL=126
Reply from 172.16.32.2: bytes=32 time=141ms TTL=126
```

Ping statistics for 172.16.32.2:

```
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
 Minimum = 125ms, Maximum = 156ms, Average = 137ms
```

PC1>ping 172.16.16.2

Pinging 172.16.16.2 with 32 bytes of data:

```
Reply from 172.16.16.2: bytes=32 time=140ms TTL=126
Reply from 172.16.16.2: bytes=32 time=156ms TTL=126
Reply from 172.16.16.2: bytes=32 time=125ms TTL=126
Reply from 172.16.16.2: bytes=32 time=141ms TTL=126
```

Ping statistics for 172.16.16.2:

```
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
```

Minimum = 125ms, Maximum = 156ms, Average = 140ms

Now displaying eigrp topology on R2 only

```
Router2#sh ip eigrp topology
IP-EIGRP Topology Table for AS 1
```

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply, r - Reply status

```
P 172.16.64.0/24, 1 successors, FD is
 20512000 via Connected, Serial2/0
P 172.16.16.0/24, 1 successors, FD is 28160
```

```
 via Connected, FastEthernet0/0
P 172.16.4.0/24, 1 successors, FD is 20512000
 via Connected, Serial3/0
P 172.16.32.0/24, 1 successors, FD is 20514560
 via 172.16.64.1 (20514560/28160), Serial2/0
P 172.16.8.0/24, 2 successors, FD is 21024000
 via 172.16.64.1 (21024000/20512000), Serial2/0
 via 172.16.4.1 (21024000/20512000), Serial3/0
```

```
Router2#sh ip eigrp neighbors
```

```
IP-EIGRP neighbors for process 1
```

| H | Address     | Interface | Hold<br>(sec) | Uptime   | SRTT<br>(ms) | RTO  | Q<br>Cnt | Seq<br>Num |
|---|-------------|-----------|---------------|----------|--------------|------|----------|------------|
| 0 | 172.16.64.1 | Se2/0     | 12            | 00:02:50 | 40           | 1000 | 0        | 20         |
| 1 | 172.16.4.1  | Se3/0     | 10            | 00:02:50 | 40           | 1000 | 0        | 24         |

## EXERCISES

### 1. What four routed protocols are supported by EIGRP?

---

---

---

---

---

### 2. When is redistribution required for EIGRP?

---

---

---

---

## **LAB SESSION 9**

### **OBJECT:**

*Studying basic LAN switch operation*

### **THEORY:**

LAN switch performs 3 operations

- Address learning
- Forward filter decision
- Loop avoidance

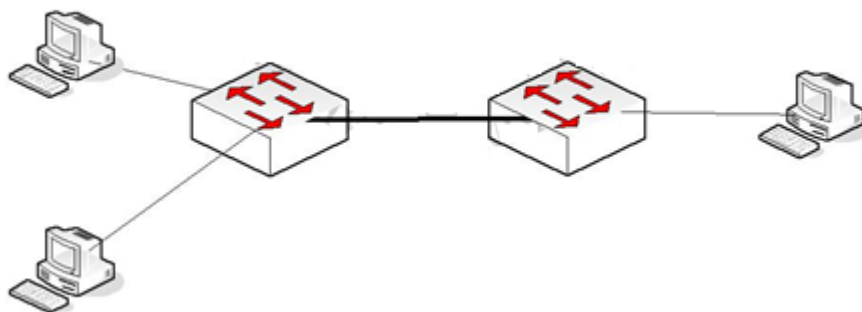
In this session, we will explore how an Ethernet switch learns addresses of the attached hosts.

#### **Address learning**

A new switch has empty MAC address table. As each frame transits switch, it learns source MAC address against the source port. As the switch does not know to which port the destination is attached, it initially transmits the frame to all ports. This process is called flooding. As the responses are received, the MAC address table is further populated.

### **PROCEDURE:**

Consider the following scenario



**Fig 9.1:** Scenario for LAN switch operation

Initially the MAC database of switch will be

```
Switch#sh mac-address-table
```

```
Mac Address Table
```

```

```

| Vlan | Mac Address    | Type    | Ports |
|------|----------------|---------|-------|
| ---- | -----          | -----   | ----- |
| 1    | 0006.2a75.100c | DYNAMIC | Fa0/1 |

Switch#

And that of second switch is;

Switch#sh mac-address-table

Mac Address Table

-----

| Vlan | Mac Address    | Type    | Ports  |
|------|----------------|---------|--------|
| ---- | -----          | -----   | -----  |
| 1    | 0060.471b.ae01 | DYNAMIC | Eth0/1 |

switch#

Now as any of the computers generates ping for any of the remaining computers, the MAC address table will grow

Switch#sh mac-address-table

Mac Address Table

-----

| Vlan | Mac Address    | Type    | Ports |
|------|----------------|---------|-------|
| ---- | -----          | -----   | ----- |
| 1    | 0006.2a75.100c | DYNAMIC | Fa0/1 |
| 1    | 0040.0ba5.183a | DYNAMIC | Fa0/1 |
| 1    | 00e0.f7a4.475c | DYNAMIC | Fa0/2 |

Switch  
#

Also for second switch:

switch#sh mac-address-table

Mac Address Table

-----

| Vlan | Mac Address    | Type    | Ports  |
|------|----------------|---------|--------|
| ---- | -----          | -----   | -----  |
| 1    | 0040.0ba5.183a | DYNAMIC | Eth1/1 |
| 1    | 0060.471b.ae01 | DYNAMIC | Eth0/1 |
| 1    | 00e0.f7a4.475c | DYNAMIC | Eth0/1 |

switch#

## **EXERCISE:**

**1. If a destination MAC address is not in the forward/filter table, what will the switch do with the frame?**

---

---

---

---

**2. If a frame is received on a switch port and the source MAC address is not in the forward/filter table, what will the switch do?**

---

---

---

---

## **LAB SESSION 10**

### **OBJECT:**

*Studying and configuring Access Lists*

### **THEORY:**

An access list is essentially a list of conditions that categorize packets. One of the most common and easiest to understand uses of access lists is filtering unwanted packets when implementing security policies. Access lists can even be used in situations that don't necessarily involve blocking packets.

There are a few important rules that a packet follows when it's being compared with an access list:

**Rule#1:** It's always compared with each line of the access list in sequential order—that is, it'll always start with the first line of the access list, then go to line 2, then line 3, and so on.

**Rule#2:** It's compared with lines of the access list only until a match is made. Once the packet matches the condition on a line of the access list, the packet is acted upon and no further comparisons take place.

**Rule#3:** There is an implicit "deny" at the end of each access list—this means that if a packet doesn't match the condition on any of the lines in the access list, the packet will be discarded. Each of these rules has some powerful implications when filtering IP packets with access lists, so keep in mind that creating effective access lists truly takes some practice.

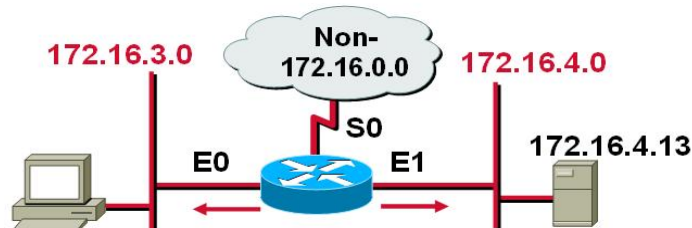
There are two main types of access lists:

1. Standard access lists
2. Extended access lists

### **Standard access lists**

These use only the source IP address in an IP packet as the condition test. All decisions are made based on the source IP address. This means that standard access lists basically permit or deny an entire suite of protocols. They don't distinguish between any of the many types of IP traffic such as web, Telnet, UDP, and so on.

Its command syntax is: `access-list <number> {permit|deny} <destination> [log]`



**Fig 10.1: Standard Access list to allow my network**



Commands on router will be

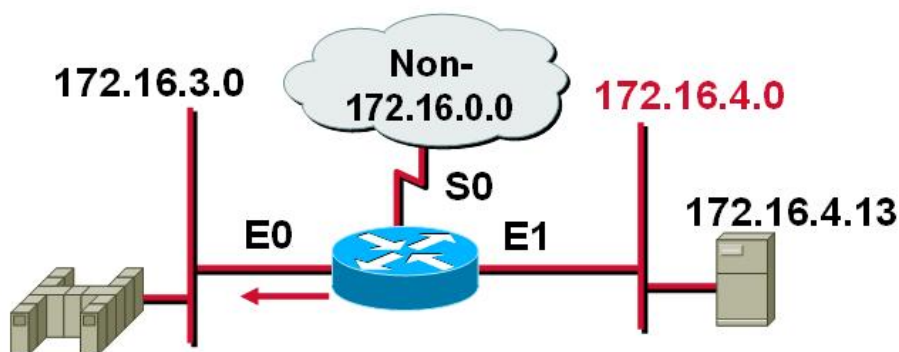
```
R1(config)#access-list 1 permit 172.16.0.0 0.0.255.255
R1(config)#interface ethernet 0
R1(config)#ip access-group 1 out
R1(config)#interface ethernet 1
R1(config)#ip access-group 1 out
```

The above commands will permit the network 172.16.0.0 only and will block other network through the router on its ethernet interfaces in its out side directions

### Extended access lists

Extended access lists can evaluate many of the other fields in the layer 3 and layer 4 headers of an IP packet. They can evaluate source and destination IP addresses, the protocol field in the Network layer header, and the port number at the Transport layer header. This gives extended access lists the ability to make much more granular decisions when controlling traffic.

Its command syntax is: access-list <number> {permit| deny}  
<protocol><source>[<ports>]<destination>[<ports>][<options>]



**Fig 10.2:** Extended access list

Commands on the router will be:

```
access-list 101 deny tcp 172.16.4.0 0.0.0.255 any eq 23
access-list 101 permit ip any any
interface ethernet 0
ip access-group 101 out
```

The above commands will deny only the Telnet from subnet 172.16.40.0 out of E0 and will permit all other traffic.

## **EXERCISE:**

**Give commands to enable logging for the given access list and to show the entries that have been blocked**

---

---

---

---

---

## **LAB SESSION 11**

### **OBJECT:**

***Introduction to Network Management Tools. Learn how Networks can be managed using these tools.***

### **THEORY:**

Network management refers to the activities, methods, procedures, and tools that can be used for maintaining following three operations on a network.

- (i) *Operation* deals with keeping the network (and the services that the network provides) up and running smoothly. It includes monitoring the network to spot problems as soon as possible, ideally before a user is affected.
- (ii) *Administration* involves keeping track of resources in the network and how they are assigned. It deals with all the “housekeeping” that is necessary to keep things under control.
- (iii) *Maintenance* is concerned with performing repairs and upgrades—for example, when a line card must be replaced, when a router needs a new operating system image with a patch, when a new switch is added to the network. Maintenance also involves corrective and preventive proactive measures such as adjusting device parameters as needed to make the managed network run “better.”
- (iv) *Provisioning* is concerned with configuring resources in the network to support a given service. For example, this might include setting up the network so that a new customer can receive voice service.

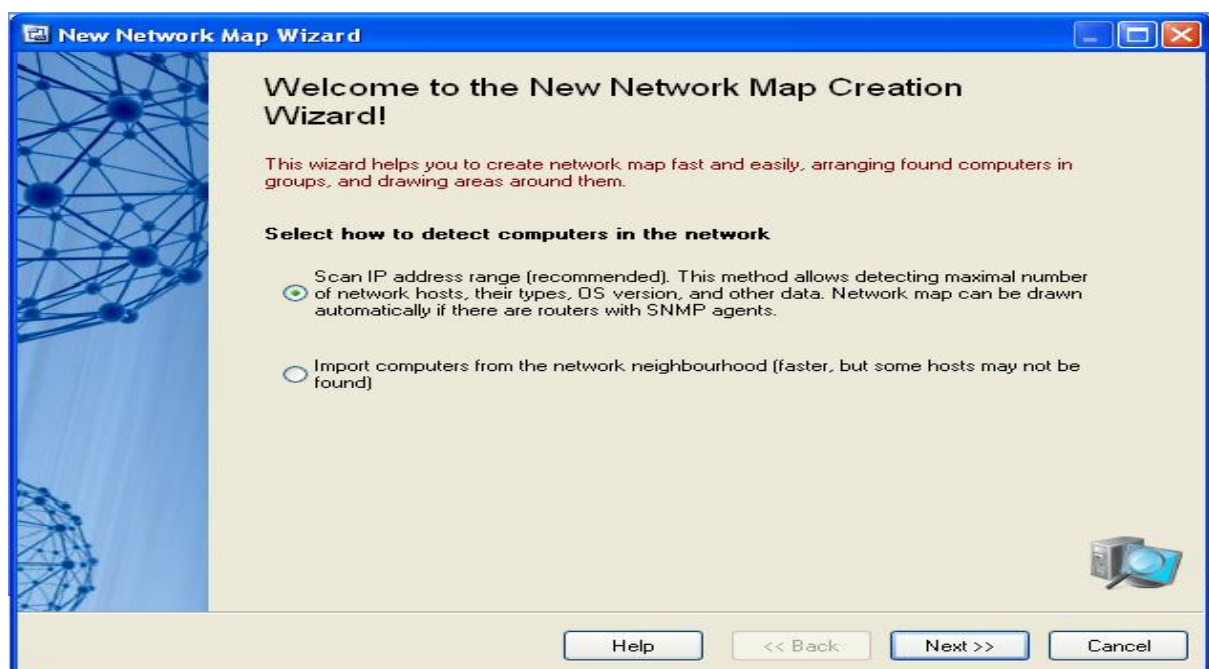
**Network Management Tools:** There are plenty of Tools that can be used for Network Management. A few of them are described below;

| Company    | Product  | URL                                                                                 | Comments                                                                                                                                                                                                    |
|------------|----------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 10-strike  | LANState | <a href="http://www.10-strike.com/lanstate/">http://www.10-strike.com/lanstate/</a> | LANState builds a network map automatically by scanning Windows network neighborhood or IP address range. It can monitor the network or individual traffic of each NE. Also supports SNMP based management. |
| Castlerock | SNMPc    | <a href="http://www.castlerock.com/">http://www.castlerock.com/</a>                 | The SNMPc 9 Network manager is appropriate for small networks It supports SNMPv3, as does the Enterprise edition that provides other capabilities. Cost of the SNMPc Enterprise and SNMPc OnLine is         |

|             |                    |                                                             |                                                                                                                                                                                                                                       |
|-------------|--------------------|-------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|             |                    |                                                             | \$12500.00 The company has been a leader in the SNMP field                                                                                                                                                                            |
| Solar Winds | Engineers Toolset  | <a href="http://solarwinds.net/">http://solarwinds.net/</a> | Provides a number of management tools ranging in price from \$145 to \$1995. The \$1495.00 package is Web-enabled. The Engineers Toolset at \$1450.00 looks like the most attractive as it contains most of the features in a package |
| MG-SOFT     | Net Inspector Lite | <a href="http://www.mg-soft.si/">http://www.mg-soft.si/</a> | Net Inspector Lite is \$495.00. MG-SOFT provides many other more comprehensive products and products can be enhanced by proxy front-end modules. There are also products that support SNMPv3                                          |

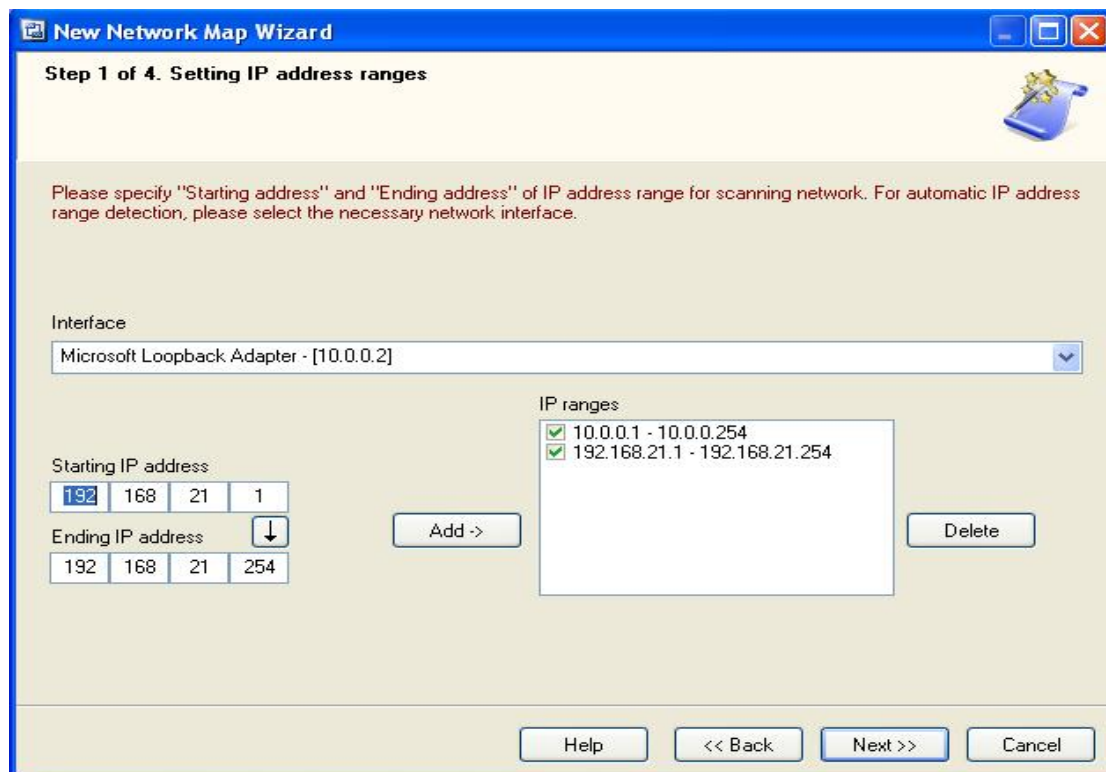
## PROCEDURE:

We will use LANSTATE tool to demonstrate how a network manager discovers a network and creates/maintains a graphical view of its network in a single window. Open LANSTATE software. Run “Map Creation Wizard” from File menu. A window will appear as shown below,



**Figure 11.1 LANState New Network Map Wizard**

Click Next, a second window like below will appear,



**Step 1 of 4. Setting IP address ranges**

Please specify "Starting address" and "Ending address" of IP address range for scanning network. For automatic IP address range detection, please select the necessary network interface.

Interface  
Microsoft Loopback Adapter - [10.0.0.2]

Starting IP address  
192 168 21 1

Ending IP address  
192 168 21 254

IP ranges

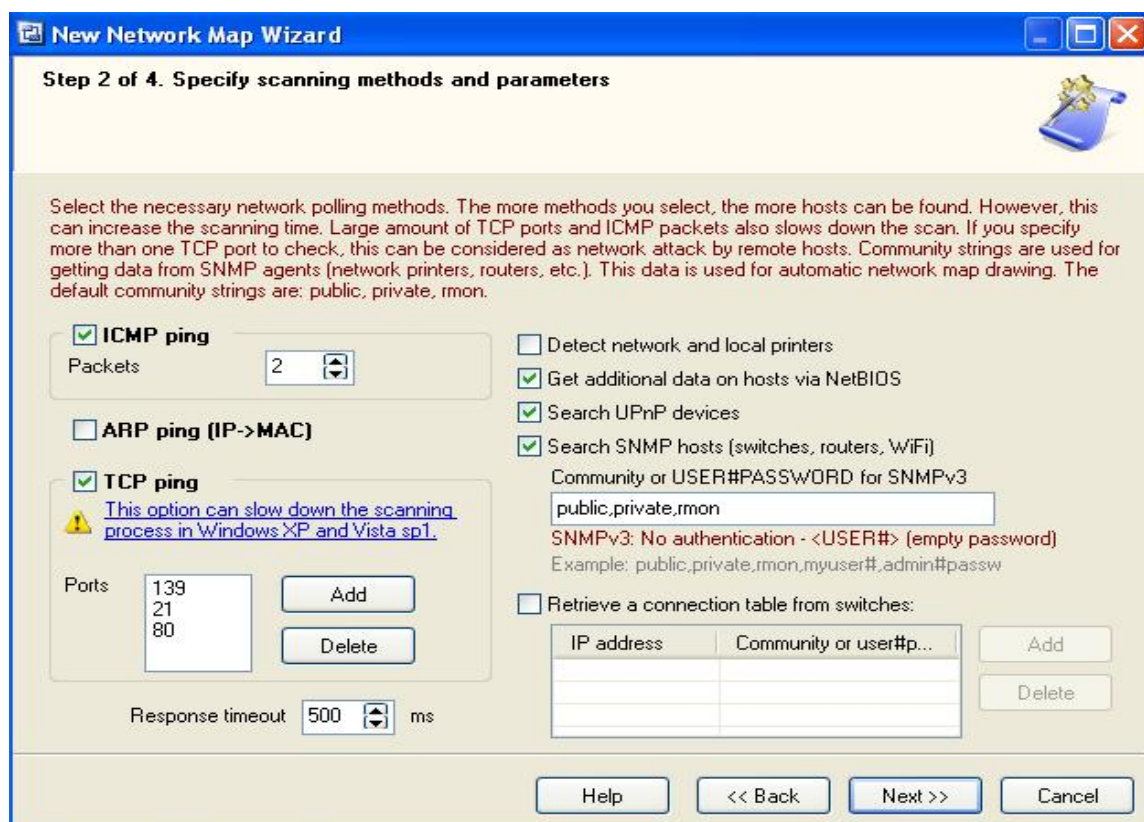
- ☒ 10.0.0.1 - 10.0.0.254
- ☒ 192.168.21.1 - 192.168.21.254

Add -> Delete

Help << Back Next >> Cancel

Figure 11.2 Setting IP Address Range on LANState

Enter the IP Address range of you target network to be discovered and click next to get following window



**Step 2 of 4. Specify scanning methods and parameters**

Select the necessary network polling methods. The more methods you select, the more hosts can be found. However, this can increase the scanning time. Large amount of TCP ports and ICMP packets also slows down the scan. If you specify more than one TCP port to check, this can be considered as network attack by remote hosts. Community strings are used for getting data from SNMP agents (network printers, routers, etc.). This data is used for automatic network map drawing. The default community strings are: public, private, rmon.

☒ ICMP ping  
Packets 2

☐ ARP ping (IP->MAC)

☒ TCP ping  
This option can slow down the scanning process in Windows XP and Vista sp1.  
Ports 139 21 80  
Add Delete  
Response timeout 500 ms

☐ Detect network and local printers

☒ Get additional data on hosts via NetBIOS

☒ Search UPnP devices

☒ Search SNMP hosts (switches, routers, WiFi)  
Community or USER#PASSWORD for SNMPv3  
public,private,rmon  
SNMPv3: No authentication - <USER#> (empty password)  
Example: public,private,rmon,myuser#,admin#passw

☐ Retrieve a connection table from switches:

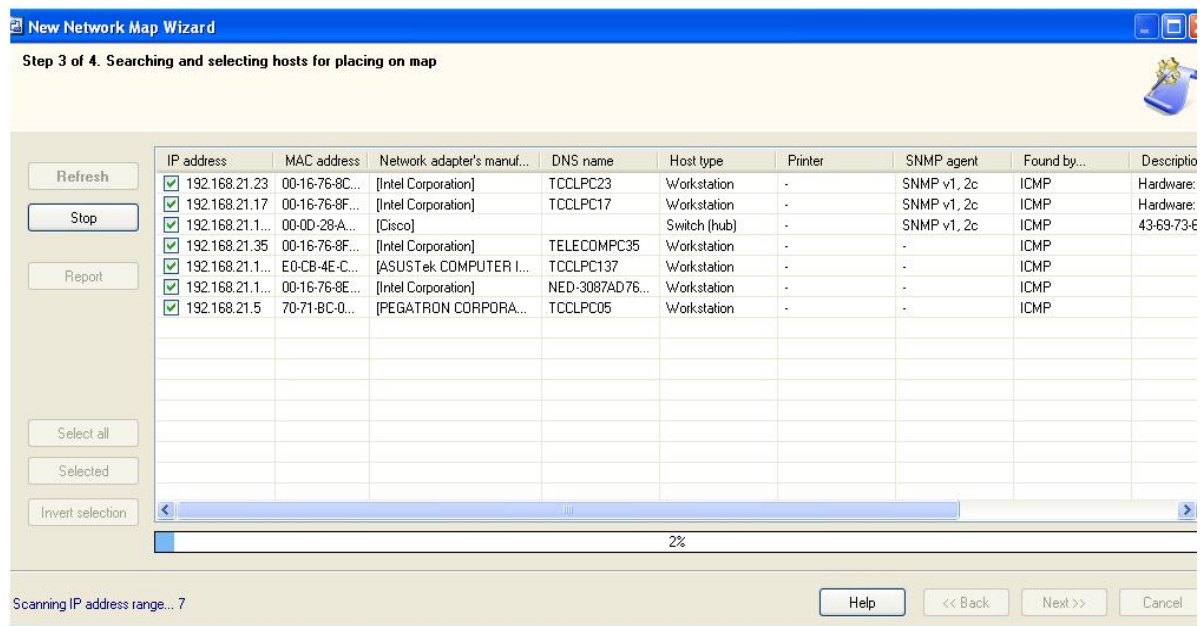
| IP address | Community or user#p... |
|------------|------------------------|
|            |                        |
|            |                        |
|            |                        |

Add Delete

Help << Back Next >> Cancel

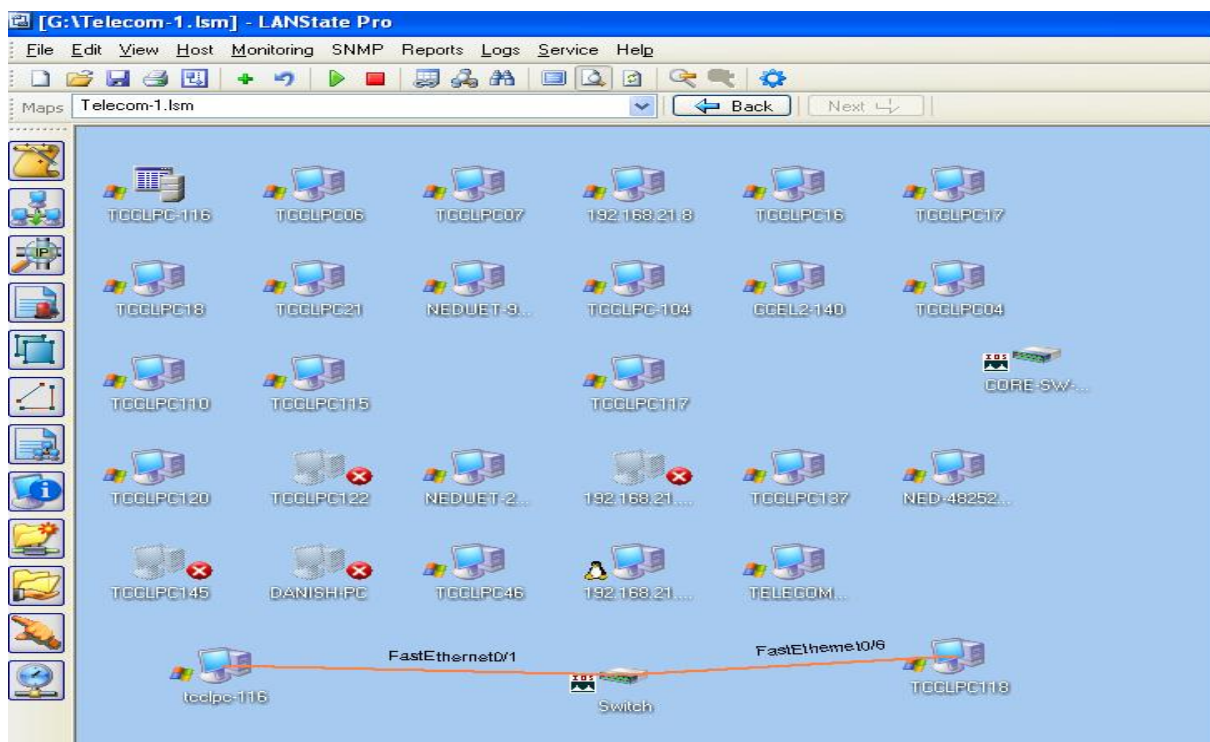
Figure 11.3 Scanning methods and parameters on LANState

Now clicking Next button will result in start of network discovery process as shown below,



**Figure 11.4 Searching and Selecting hosts for placing on map**

The finally discovered network would be like as shown below,

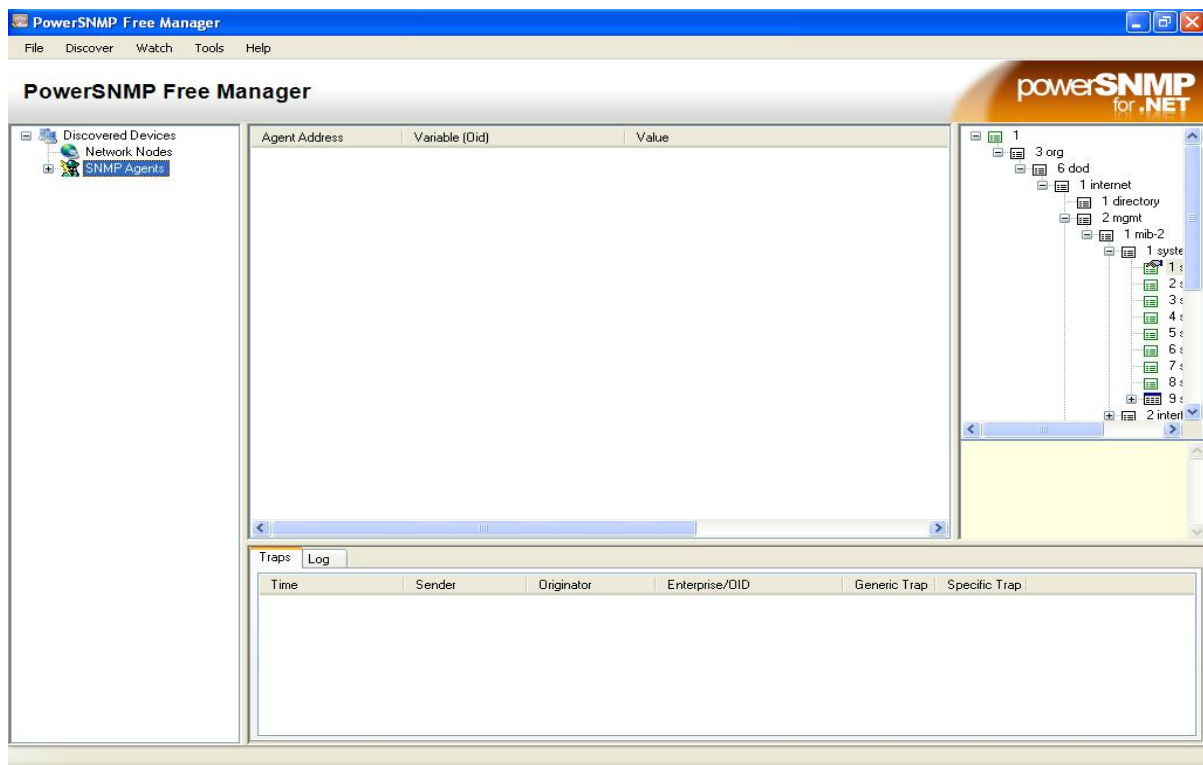


**Figure 11.5 LANState Network Map**

Students are encouraged to explore the features of this software.

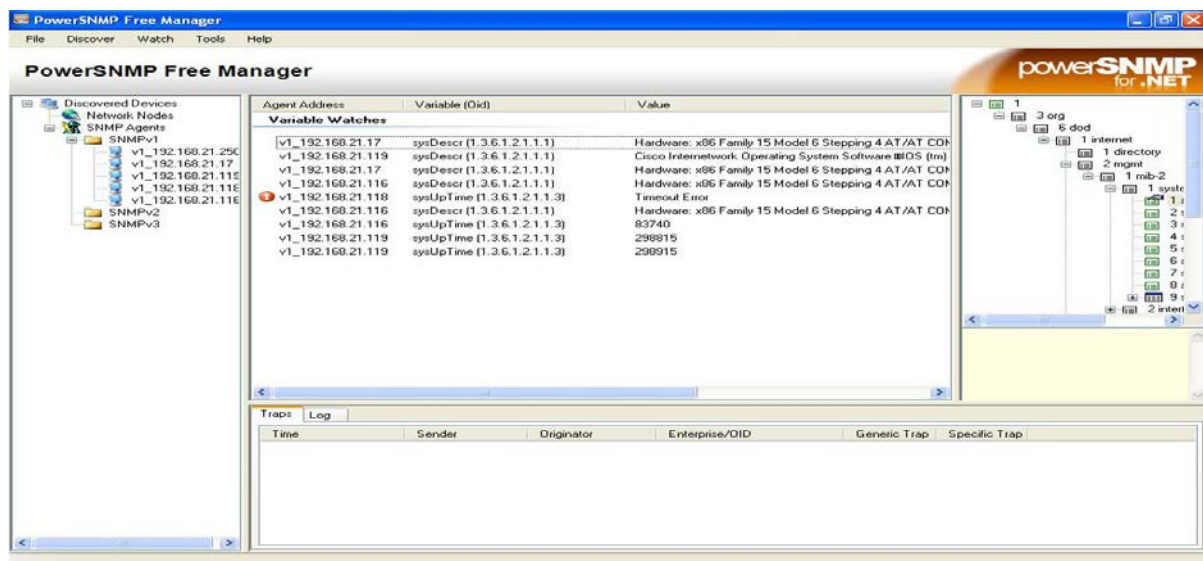


Now open the PowerSnmp from start menu. You should get a window like below,



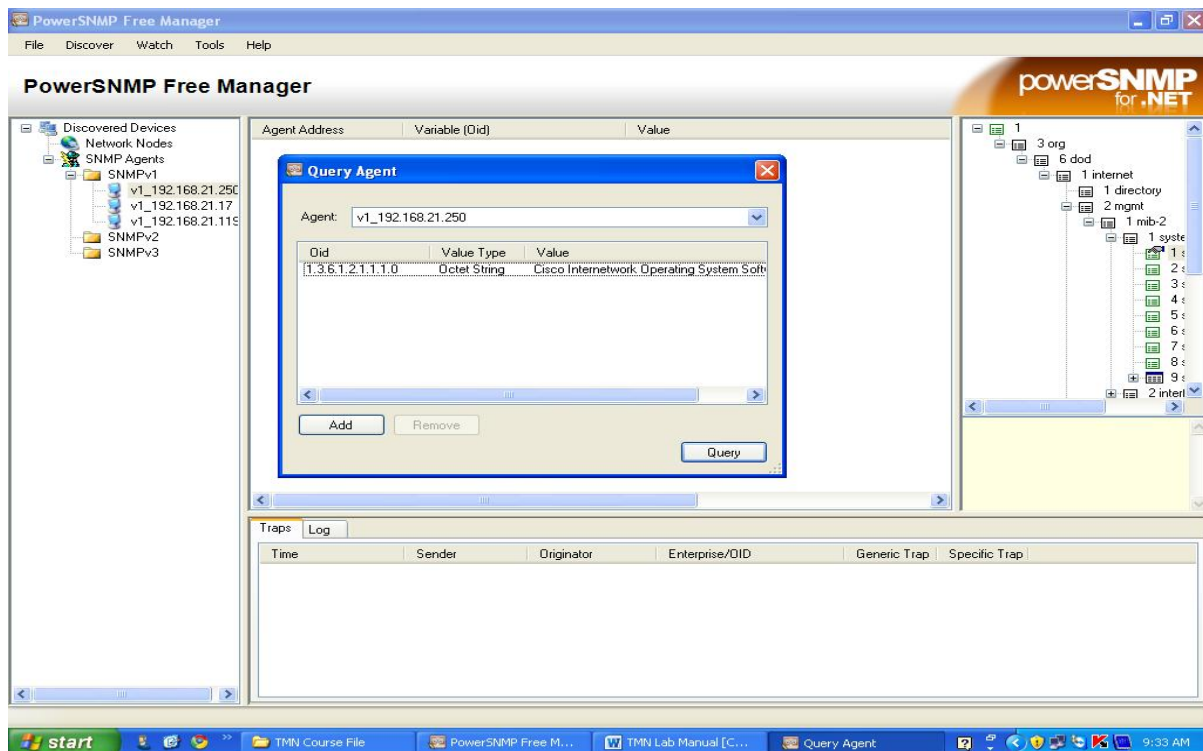
**Figure 11.6 PowerSNMP Free Manager Main window**

Go to Discover → SNMP Agents to obtain the following window, In the address bar you can specify the target network to be discovered or use the default broadcast address and press find. Based on the SNMP community (public and NED) set in the properties clients with SNMP agent enabled will be discovered. Add these discovered clients to obtain the below window,



**Figure 11.7 Network Agent Discovery**

Now select the parameter of sysDescr from the SNMP MIT shown in the rightmost window and select any discovered agent from left most window, right click on it and select query, a windows will pop up in which press query button again to obtain the below window .You can see that it has returned the complete description of the selected client. You can similarly any of the supported parameter in the SNMP MIT.



**Figure 11.8 Query Agent**

Now select any of the SNMP agent and select the sysUptime parameter from MIT and select the add watch option from the right click options on the selected agent.

## EXERCISE:

- 1) Design a network in packet tracer 5.3. Add a server to the network and configure HTTP service on it. Use this service from any client in the network.
- 2) Using LANSTATE discover a network (preferably your home network if applicable) and export the discovered network to Microsoft Visio.



## **LAB SESSION 12**

### **OBJECT:**

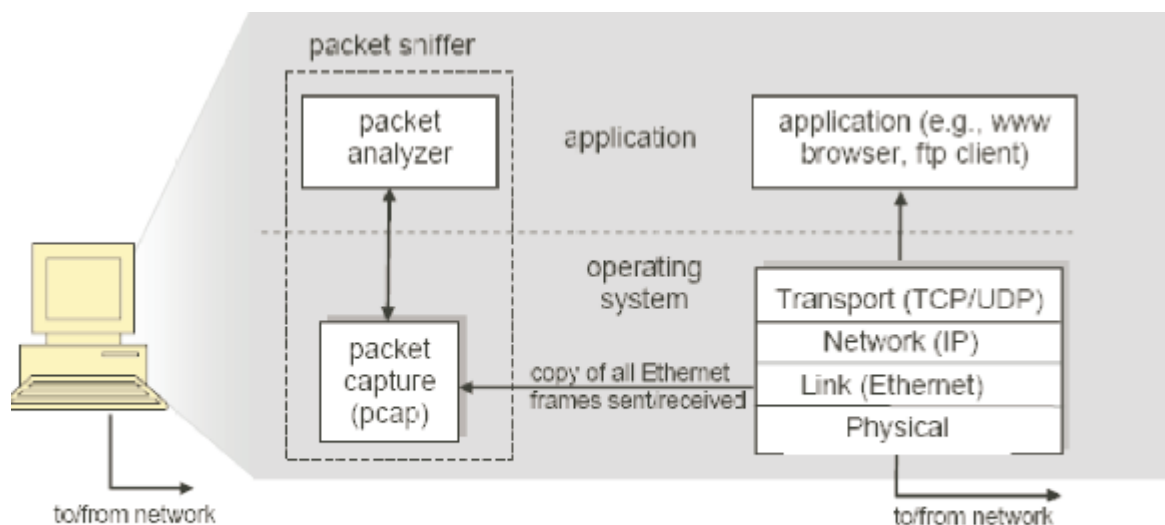
***Introduction to WireShark (Network Protocol Analyzer /Packet Sniffer) and Live network monitoring***

### **THEORY**

A better way to understand network protocols is to observe how they actually work. A basic tool for observing the messages exchanged between executing protocol entities is the packet sniffer, which is an essential part of network protocol analyzer. WireShark is a free and open-source network protocol analyzer that runs on various operating systems including Linux, Unix, Mac, and Windows.

#### **WireShark:**

WireShark (previously called Ethereal) is one of the most widely used network protocol analyzer. It passively sniffs packets that are sent from or received by a designated network interface, but never sends packets itself. It receives a copy of packets that are sent from or received by the applications and protocols executing on the end-system (e.g., your computer). WireShark also has a graphical front-end to display the packets that it sniffs.



**Figure 12.1 Structure of Packet Capture Software/ Protocol Analyzer**

Figure 12.1. shows the structure of a network protocol analyzer. At the right of the figure shows the protocol stack and applications (such as a web browser or an FTP client) that normally run on your computer. The network protocol analyzer, shown within the dashed rectangle, has two parts, the packet capture and the packet analyzer. The packet capture library receives a copy of every link-layer frame that is sent from or received by a designated network interface. The messages exchanged by higher layer protocols such as HTTP, FTP, TCP, UDP, DNS, or IP all are eventually encapsulated in link-layer frames that are

transmitted over physical media such as an Ethernet cable. In Figure 12.1 the assumed physical media is an Ethernet, and so all upper layer protocols' headers are eventually encapsulated within an Ethernet frame. Capturing all link-layer frames thus gives you all messages sent from or received by all protocols and applications executing in your computer.

The second component is the packet analyzer, which displays the contents of all fields within a link-layer frame. In order to do so, the packet analyzer must understand the structure of messages exchanged by the protocols. For example, we are interested in displaying the various fields in messages exchanged by the HTTP protocol in Figure 12.1. The packet analyzer understands the format of Ethernet frames, and so it can identify the IP datagram within an Ethernet frame. It also understands the IP datagram format, so it can extract the TCP segment within the IP datagram. It understands the TCP segment structure, so it can extract the HTTP message contained in the TCP segment. Finally, it understands the HTTP protocol and so, for example, knows that an HTTP message may contain the string of "GET", "POST" or "HEAD".

### **Layered Protocol:**

Two reference models are used to describe the network architecture, the OSI/ISO reference model and the TCP/IP reference model. The OSI/ISO model divides the network into seven layers and the TCP/IP model divides the network into four layers. No matter which model is used, the basic principle of the layered architecture is that each layer performs some services for the layer above it.

## **PROCEDURE**

### **Installation**

Wireshark is free to download at <http://www.wireshark.org/>. How to build and install Wireshark onto machines first consult the Wireshark User's Guide, the Wireshark Developers Guide and the various README files provided with Wireshark.

### **Starting Wireshark**

When you run Wireshark, you will see the graphical user interface (GUI) as shown in Figure 12.2. There are four main fields:

- a. **Filter field:** Used to filter out uninteresting packets with the entered specifications, so you can choose which packets should (not) be shown on the screen;
- b. **Captured packets:** Lists the packets captured by the selected interface;
- c. **Details of selected packet:** Lists information about the packet that is selected in the captured packets window;
- d. **Content of packet in hex/ASCII:** Displays the content of the captured packet, in hex and ASCII.

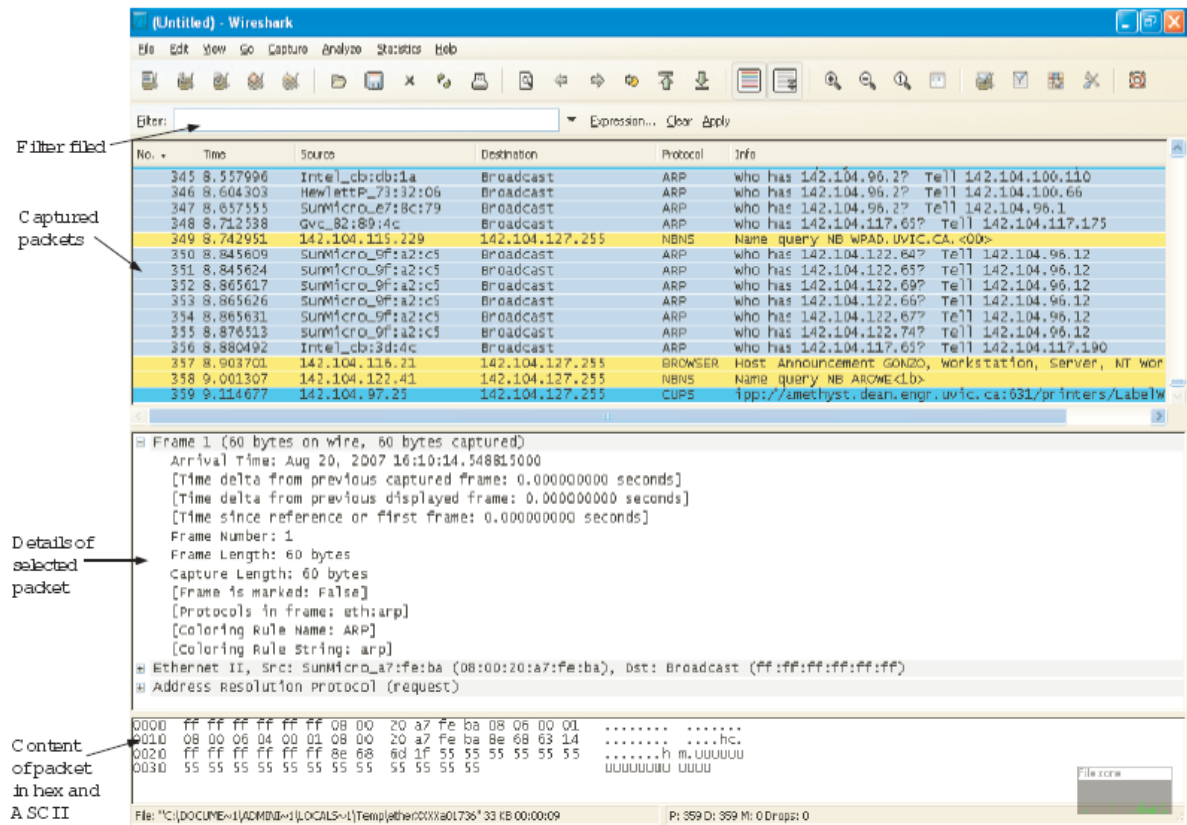


Figure 12.2: Wireshark Graphical User Interface

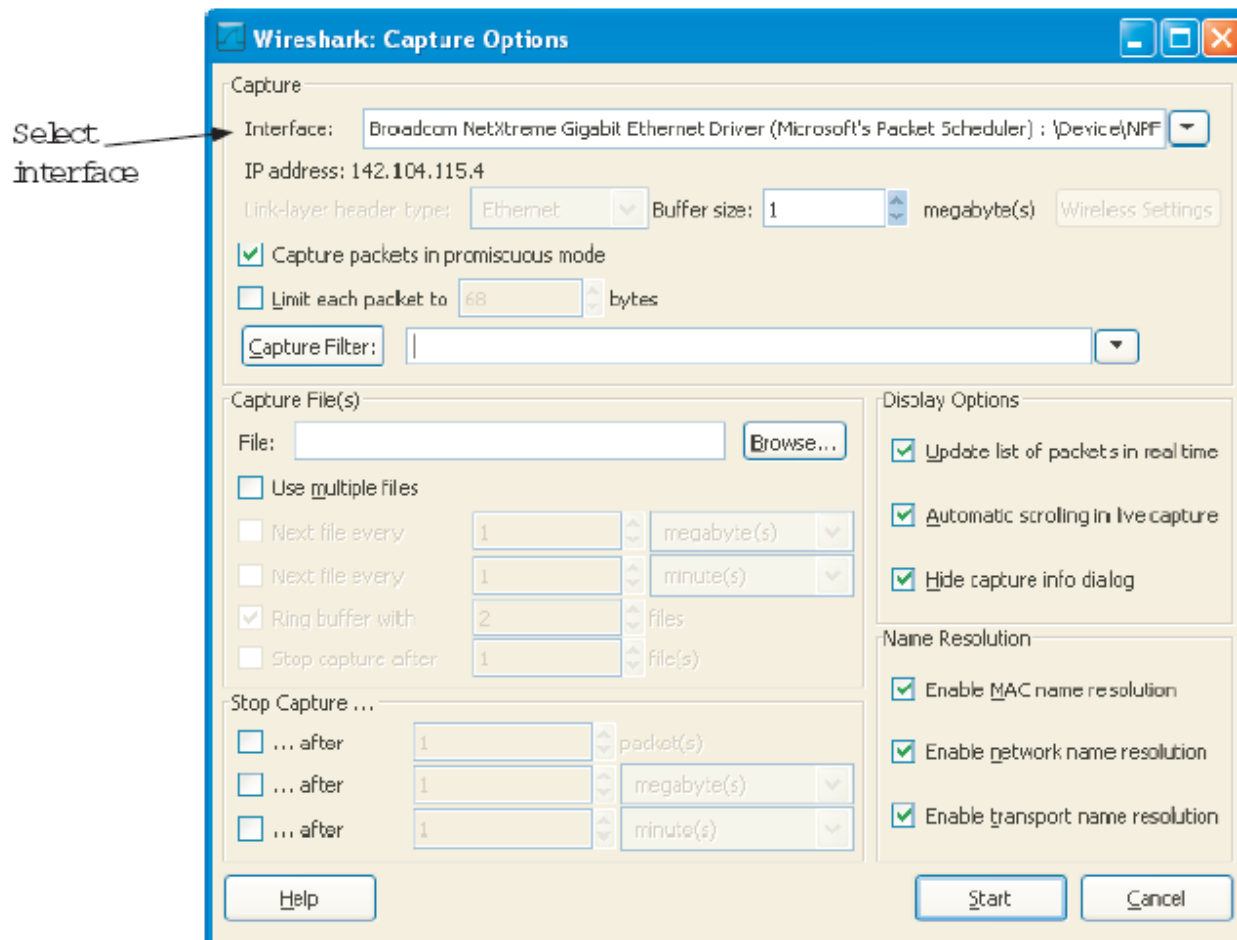
### Capture Trace

Use the following procedure to capture the trace:

- Pick a URL and fetch it through web browser. For example, open a new window of your browser and type <https://www.google.com.pk>.
- Close web browser. By minimizing browser activity you will stop your computer from fetching unnecessary web content, and avoid incidental traffic in the trace.
- Now launch Wireshark. Choose the network interface that we would like to capture the packets on. To do this, select “Capture options” from the command menu. Select the interface you are using. Uncheck “Capture packets in promiscuous mode”. This mode is useful to overhear packets sent to/from other computers on broadcast networks. We only want to record packets sent to/from your computer. See figure 12.3
- Use capture filter “tcp port 80”. This filter will record only standard web traffic and not other kinds of packets that your computer may send. Click “Start” to start the packet capture process.
- When the capture is started, repeat the web fetch using web browser. This time, the packets will be recorded by Wireshark as the content is transferred.
- After the fetch is successful, return to Wireshark and use the menus or buttons to stop the trace (“Capture Stop”). If you have succeeded, the upper Wireshark window will show multiple packets. How many packets being captured will depend on the size of the web page. An example is shown in Figure 12.4

## Layered Protocol

By inspecting the captured trace, we can understand the layered protocol.



**Figure 12.3 Capture options window**

- Select an HTTP GET packet. This packet carries the HTTP request sent from your computer to the server.
- HTTP is the application layer web protocol used to fetch URLs. It runs on top of the TCP/IP transport and network layer protocols. The link layer protocol shown in the figure is Ethernet. It may be other protocol, depends on your network.
- Click on one HTTP packet, and turn to the middle panel with details of the packet. The first block is "Frame". This is a record that describes overall information about the packet, including when it was captured and how many bits long it is.
- The second block is "Ethernet" (You may have taken trace in a computer with 802.11, but still you will see an Ethernet block. This is because Wireshark capture traffic in Ethernet format determined on the capture options. See Link-layer header type.).
- Then we can see IP, TCP, and HTTP. This is a bottom-up order, because as packets are passed down the protocol stack, the header of the lower layer protocol is added to the front of the information from the higher layer protocol. That is, the lower layer protocols come first in the packet.

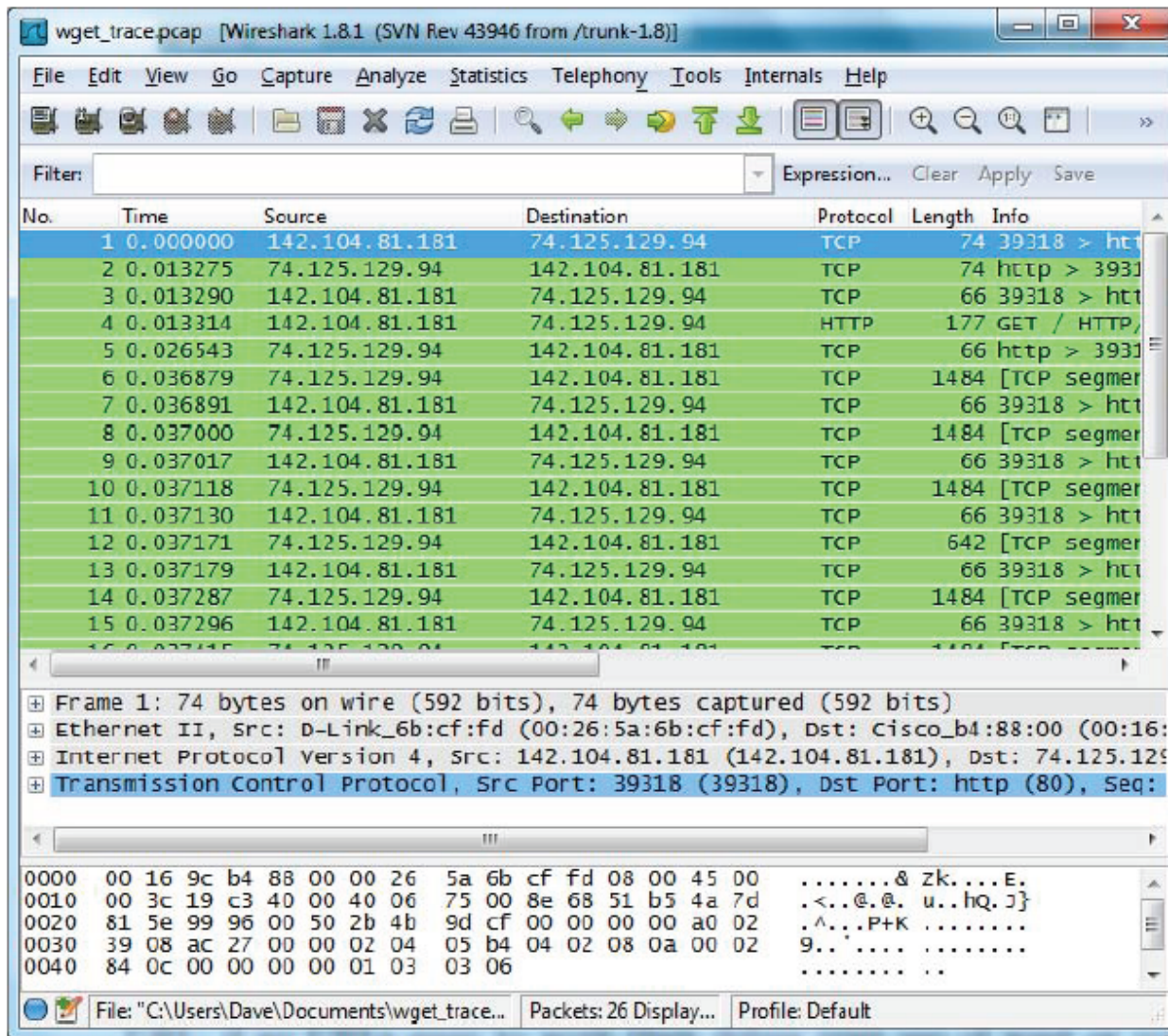


Figure 12.4 Packet trace

When an Ethernet frame arrives at a computer, the Ethernet layer must hand the packet that it contains to the next higher layer to be processed. In order to do this, the protocol use information in its header to determine the higher layer data unit encapsulated.

### EXERCISE:

Capture a SNMP packet using Wireshark and record your observation.

## **LAB SESSION 13**

### **OBJECT:**

- *To study the concept of a Network Monitoring System (NMS)*
- *To study a SNMP based NMS and observe communication between a managed element (CISCO 2950 Catalyst Switch) and NMS system*

### **THEORY**

Short for Network Management System, NMS is a computer that has been setup to monitor and/or manage a Data/voice network and the devices contained in that network. Network management refers to the activities, methods, procedures, and tools that pertain to the operation, administration, maintenance, and provisioning of networked systems.

#### **NMS implementation using SNMP**

Simple Network Management Protocol (SNMP) is a UDP-based network protocol. It is used mostly in network management systems to monitor network-attached devices for conditions that warrant administrative attention. SNMP is a component of the Internet Protocol Suite as defined by the IETF. It consists of a set of standards for network management, including an application layer protocol, a database schema, and a set of data objects.

SNMP exposes management data in the form of variables on the managed systems, which describe the system configuration. These variables can then be queried (and sometimes set) by managing applications.

In typical SNMP use, one or more administrative computers have the task of monitoring or managing a group of hosts or devices on a computer network. Each managed system (also called Slave) executes, at all times, a software component called an agent (see below) which reports information via SNMP to the managing systems (also called Masters).

Essentially, SNMP agents expose management data on the managed systems as variables (such as "free memory", "system name", "number of running processes", "default route"). But the protocol also permits active management tasks, such as modifying and applying a new configuration. The managing system can retrieve the information through the GET, GETNEXT and GETBULK protocol operations or the agent will send data without being asked using TRAP or INFORM protocol operations. Management systems can also send configuration updates or controlling requests through the SET protocol operation to actively manage a system. Configuration and control operations are used only when changes are needed to the network infrastructure. The monitoring operations are usually performed on a regular basis.

The variables accessible via SNMP are organized in hierarchies. These hierarchies such as type and description of the variable are described by Management Information Bases (MIBs). Typically, SNMP uses UDP ports 161 for the agent and 162 for the manager. The manager may send requests from any available source port to port 161 in the agent. The agent response will be sent back to the source port. The manager typically receives notifications on



port 162. The agent may generate notifications from any available port.

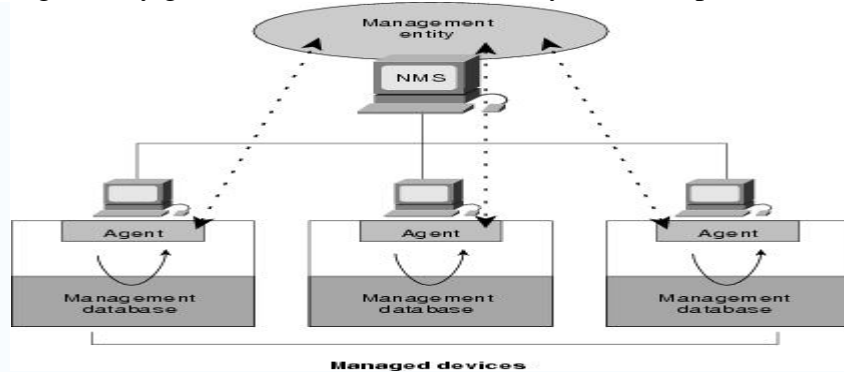


Figure 13.1 Two-tier model

## PROCEDURE

We have a very simple network comprising of following components;  
 A CISCO 2950 CATALYST Switch configured for SNMP  
 A PC loaded with IReasoning MIB Browser.

We have simple Ethernet connectivity between the two elements with following IPs assigned:

192.168.10.1 assigned to PC

192.168.10.2 assigned on VLAN1 in Cisco 2950 Switch.

Open the IReasoning MIB Browser on the PC. It should load as shown below;

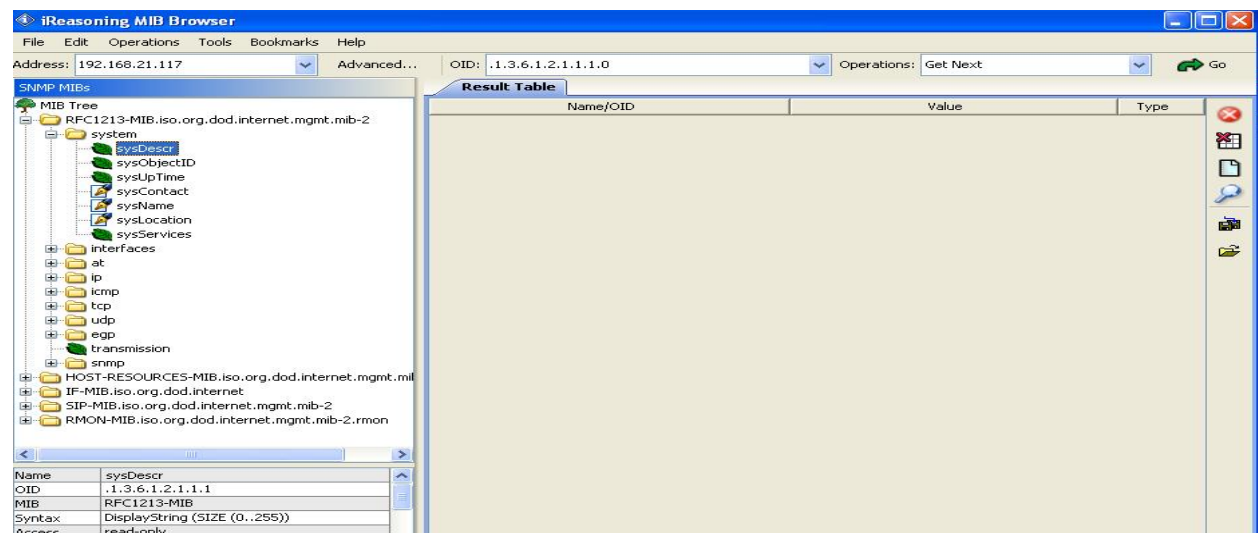


Figure 13.2 MIB Browser window

Ensure that RFC-1213 MIB file is loaded in the left pane. Also go to 'Advanced' Option and ensure that the SNMP v1 is selected, the Port No is 161 and the community name is 'public'. Now go to the CISCO Switch 2950 prompt using Hyper Terminal and type following commands to configure SNMP on the Switch:

Switch#conf t

Switch(config)#snmp-server enable

Switch(config)#snmp-server enable informs

Switch(config)#snmp-server community public

Now go back to IReasoning MIB Browser and go to RFC-1213→Interfaces→ifTable in the left pane and right click on it and then select the option Table View. You will get the following view:

| #Index | IFDescr          | IFType         | IFMTU | IFSpeed    | IFPhysAddress     | IFAdminS... | IFOperStatus | IFLastChange        |     |
|--------|------------------|----------------|-------|------------|-------------------|-------------|--------------|---------------------|-----|
| 1      | FastEthernet0/1  | ethernetCsmacd | 1500  | 100000000  | 00-0D-28-AC-76-81 | up          | up           | 30 minutes 31 se... | 143 |
| 2      | FastEthernet0/2  | ethernetCsmacd | 1500  | 100000000  | 00-0D-28-AC-76-82 | up          | up           | 22 seconds          | 995 |
| 3      | FastEthernet0/3  | ethernetCsmacd | 1500  | 100000000  | 00-0D-28-AC-76-83 | up          | down         | 17 seconds          | 64  |
| 4      | FastEthernet0/4  | ethernetCsmacd | 1500  | 100000000  | 00-0D-28-AC-76-84 | up          | down         | 17 seconds          | 64  |
| 5      | FastEthernet0/5  | ethernetCsmacd | 1500  | 100000000  | 00-0D-28-AC-76-85 | up          | down         | 17 seconds          | 64  |
| 6      | FastEthernet0/6  | ethernetCsmacd | 1500  | 100000000  | 00-0D-28-AC-76-86 | up          | down         | 17 seconds          | 64  |
| 7      | FastEthernet0/7  | ethernetCsmacd | 1500  | 100000000  | 00-0D-28-AC-76-87 | up          | down         | 17 seconds          | 64  |
| 8      | FastEthernet0/8  | ethernetCsmacd | 1500  | 100000000  | 00-0D-28-AC-76-88 | up          | down         | 17 seconds          | 64  |
| 9      | FastEthernet0/9  | ethernetCsmacd | 1500  | 100000000  | 00-0D-28-AC-76-89 | up          | down         | 17 seconds          | 64  |
| 10     | FastEthernet0/10 | ethernetCsmacd | 1500  | 100000000  | 00-0D-28-AC-76-8A | up          | down         | 17 seconds          | 64  |
| 11     | FastEthernet0/11 | ethernetCsmacd | 1500  | 100000000  | 00-0D-28-AC-76-8B | up          | down         | 17 seconds          | 64  |
| 12     | FastEthernet0/12 | ethernetCsmacd | 1500  | 100000000  | 00-0D-28-AC-76-8C | up          | down         | 17 seconds          | 64  |
| 13     | FastEthernet0/13 | ethernetCsmacd | 1500  | 100000000  | 00-0D-28-AC-76-8D | up          | down         | 17 seconds          | 64  |
| 14     | FastEthernet0/14 | ethernetCsmacd | 1500  | 100000000  | 00-0D-28-AC-76-8E | up          | down         | 17 seconds          | 64  |
| 15     | FastEthernet0/15 | ethernetCsmacd | 1500  | 100000000  | 00-0D-28-AC-76-8F | up          | down         | 17 seconds          | 64  |
| 16     | FastEthernet0/16 | ethernetCsmacd | 1500  | 100000000  | 00-0D-28-AC-76-90 | up          | down         | 17 seconds          | 64  |
| 17     | FastEthernet0/17 | ethernetCsmacd | 1500  | 100000000  | 00-0D-28-AC-76-91 | up          | down         | 17 seconds          | 64  |
| 18     | FastEthernet0/18 | ethernetCsmacd | 1500  | 100000000  | 00-0D-28-AC-76-92 | up          | down         | 17 seconds          | 64  |
| 19     | FastEthernet0/19 | ethernetCsmacd | 1500  | 100000000  | 00-0D-28-AC-76-93 | up          | down         | 17 seconds          | 64  |
| 20     | FastEthernet0/20 | ethernetCsmacd | 1500  | 100000000  | 00-0D-28-AC-76-94 | up          | down         | 17 seconds          | 64  |
| 21     | FastEthernet0/21 | ethernetCsmacd | 1500  | 100000000  | 00-0D-28-AC-76-95 | up          | down         | 17 seconds          | 64  |
| 22     | FastEthernet0/22 | ethernetCsmacd | 1500  | 100000000  | 00-0D-28-AC-76-96 | up          | down         | 17 seconds          | 64  |
| 23     | FastEthernet0/23 | ethernetCsmacd | 1500  | 100000000  | 00-0D-28-AC-76-97 | up          | down         | 17 seconds          | 64  |
| 24     | FastEthernet0/24 | ethernetCsmacd | 1500  | 100000000  | 00-0D-28-AC-76-98 | up          | down         | 17 seconds          | 64  |
| 25     | Null0            | other          | 1500  | 4294967295 |                   | up          | up           | 0 millisecond       | 0   |
| 26     | Vlan1            | propVirtual    | 1500  | 1000000000 | 00-0D-28-AC-76-80 | up          | up           | 53 seconds          | 128 |

Figure 13.3 ifTable snapshot

**EXERCISE:** Observe the different parameters of the managed element being shown in this Table and write your comments in the result section



## **LAB SESSION 14**

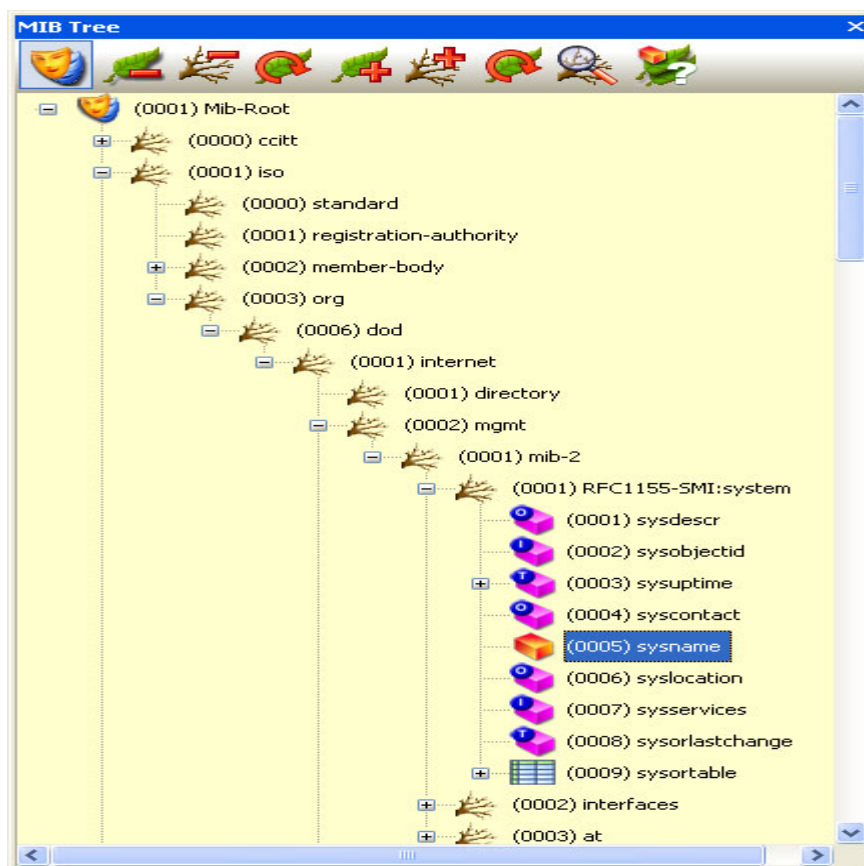
### **OBJECT:**

*To study the MIB File structure based on RFC-1155 SMI and RFC 1212.*

### **THEORY**

#### **Introduction**

The MIB are files describing the objects used by the SNMP protocol. The MIB term stands for Management Information Base because the structure of it is quite similar to a database description. This is a text file following the ASN1 standard. The RFC 1155 defines writing rules of the MIB file in SMI V1 and the RFC 1213 contains the object definition that should be implemented in an agent. MIB are organized in hierarchy that looks like a tree. The structure of this tree follows standard defined by RFC (Request For Comments). Currently there are two versions, the SMI V1 and the V2. The MIB tree representation in any MIB Compiler like LorientPro with snmp object name and mib file name.



#### **SNMP Versions and Definitive Documents**

SNMP is defined by IETF (<http://www.ietf.org>) through a group of RFCs shown below.

rfc1155 : Structure and Identification of Management Information for TCP/IP based internets  
rfc1156 : Management Information Base Network  
rfc1157 : A Simple Network Management Protocol  
rfc1441 : Introduction to SNMP v2  
rfc2579 : Textual Conventions for SNMP v2  
rfc2580 : Conformance Statements for SNMP v2  
rfc2578 : Structure of Management Information for SNMP v2  
rfc3416 : Protocol Operations for SNMP v2  
rfc3417 : Transport Mappings for SNMP v2  
rfc3418 : Management Information Base for SNMP v2  
rfc3410 : Introduction and Applicability Statements for Internet Standard Management Framework  
rfc3411 : Architecture for Describing SNMP Frameworks  
rfc3412 : Message Processing and Dispatching for the SNMP  
rfc3413 : SNMP Applications  
rfc3414 : User-based Security Model (USM) for SNMP v3  
rfc3415 : View-based Access Control Model for the SNMP  
rfc3584 : Coexistence between SNMP v1, v2 and v3

### **Proprietary MIB**

The proprietary MIB are attached to the private branch in the MIB tree and are defined and registered by each constructor. Each constructor should ask for an enterprise number allowing him to insert his MIB entries in the global tree.

LoriotPro owns the 7291 number and then has its proper MIB file.

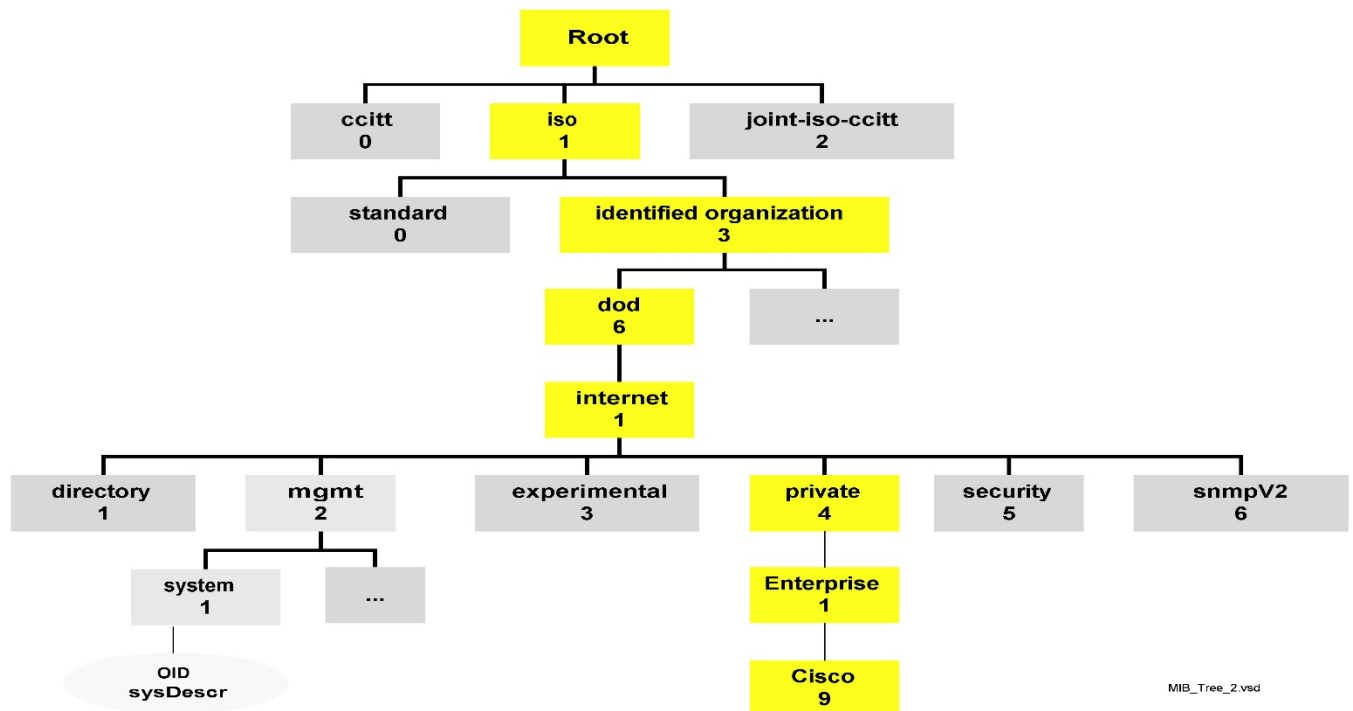
All proprietary MIB are query by the path

**iso(1).org(3).dod(6).internet(1).private(4).enterprises.xxx**

Here an example of the private MIB form the Cisco Company.



The assigned number to Cisco MIB is 9 and fit in the tree like shown hereafter:



The SMI V2 norm defines in the RFC 1902 modifies the syntax of the object definition field.

Example : SysName object definition in SMI V1

```

sysName OBJECT-TYPE
 SYNTAX DisplayString (SIZE (0..255))
 ACCESS read-write
 STATUS mandatory
 DESCRIPTION
 "An administratively-assigned name for this managed node. By convention,
 this is the node's
 fully-qualified domain name."
 ::= { system 5 }

```

The Sysname object is attached to the upper tree object System with index number 5. By taking each consecutive object definition, it is possible to walk the tree up to the root. The definitions of SNMP objects that are nodes in the tree representation use the keyword OBJECT IDENTIFIER and not OBJECT-TYPE like leaf objects.

```

system OBJECT IDENTIFIER ::= { mib-2 1 }
The system object is linked to the mib-2 object with index 1
mib-2 OBJECT IDENTIFIER ::= { mgmt 1 }
The mib-2 object is linked to the mgmt object with index 1
mgmt OBJECT IDENTIFIER ::= { internet 2 }
The mgmt object is linked to the internet object with index 2
internet OBJECT IDENTIFIER ::= { dod 1 }

```

The **internet** object is linked to the **dod** object with index 1

dod OBJECT IDENTIFIER ::= { org 6 }

The **dod** object is linked to the **org** object with index 6

org OBJECT IDENTIFIER ::= { iso 3 }

The **org** object is linked to the **iso** object with index 3

This gives in the numerical format: **1.3.6.1.2.1.1.5.0**

### **EXERCISE:**

**You are required to design your own MIB file for a customized information model and place it as a group in at suitable location in the MIT. Your MIB file should be successfully loaded with MIB Browser. Attach the screen shot of the loaded MIB file in IReasoning Mib Browser.**

## **LAB SESSION 15**

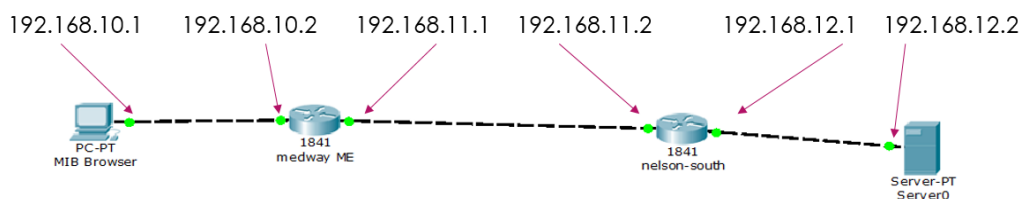
### **OBJECT:**

*To design a SNMP based managed network in Packet Tracer 5.3 using SNMPv2c and RW and Read only communities.*

**THEORY** Covered in class theory

### **PROCEDURE:**

Design the following network in the packet Tracer and assign names as shown below.



**Figure 15.1 Network Topology**

The medway-ME router is configured to serve an SNMP community medway-public for read-only access, and medway-private for read-write access.

Using SNMP it is possible to retrieve and set various configuration options.

#### **Test SNMP Requests from medway-ME router**

a. Start the MIB browser from the desktop on MIB\_Browser PC. Click the Advanced button and use the following details:

Address: 192.168.10.2

Port: 161

Read Community: medway-public

Write Community: medway-private

SNMP Version: v3

Select OK

b. Suppose we wanted to view the IOS version on the medway-ME router, we need to select this parameter from the MIB options.

Expand the MIB tree on the left side of the application to select the node with the name:

MIB Tree.router\_std MIBs.iso.org.dod.internet.mgmt.mib-2.system.sysDescr

Make a note of the OID value.

c. Ensure that Get is selected in the Operations field of the MIB browser. Press the GO button to retrieve the value from the device.

Which version of IOS is running? Verify this directly on the router via the CLI - type show version at the priv exec prompt.

d. Now select the node with the name:

MIB Tree.router\_std MIBs.iso.org.dod.internet.mgmt.mib-2.system.sysUpTime

Make a note of the OID value.

e. Ensure that Get is selected in the Operations field of the MIB browser. Press the GO button to retrieve the value from the device.

How long has the router been 'up'?

f. Now get the value of .sysName node.

What is its value? Make a note of the OID value.

Compare the OID values. What is their purpose?

### **Setting Variables**

Some variables can be set from the management station. These have an Access value of read-write.

When a piece of managed hardware is installed onto a network, it may be necessary to add some information peculiar to your organization.

g. Select the .sysContact node and change the operation to Set.

Select the OctetString data option and enter your name.

Press GO.

h. Change the operation back to Get and press GO again to verify that the data was set correctly.

i. Set the .sysLocation node value to ME4 4TB and verify that it has been set correctly.

### **Using Get Bulk**

The Get Bulk command can be used to request several parameters at once instead of requesting them singly.

j. Expand the MIB tree on the left side of the application to select the node with the name:

MIB Tree.router\_std MIBs.iso.org.dod.internet.mgmt.mib-2.interfaces

Ensure that Get Bulk is selected in the Operations field of the MIB browser. Press the GO button to retrieve the values from the device.

Note that now you have received all of the data held under the interfaces branch of the MIB tree.

How many interfaces are there? Which of the three interfaces is administratively down?

### Accessing other devices

Now click the Advanced button and use the following details:

Address: 192.168.11.2

Port: 161

Read Community: medway-public

Write Community: medway-private

SNMP Version: v3

Select OK

You can now access the nelson-south router

k. Access the .sysLocation node and give this an appropriate value. Remember to use the OctetString for the Type.

l. Access the .sysContact node and give this an appropriate value. Remember to use the OctetString for the Type.

m. How many interfaces does this router have in its MIB table under Tree.router\_std MIBs.iso.org.dod.internet.mgmt.mib-2.interfaces

Check the exchange of PDUs via the Simulation tab on Packet Tracer. View the contents of a request PDU and a Response PDU.

### EXERCISE:

Design your own network like above in packet tracer and repeat the above steps.

---

---

---

---

---

---

---

This image shows a single sheet of white paper with horizontal blue or grey ruling lines, typical of notebook paper. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.