Batch: A2 Roll No.: 16010121045

Experiment / assignment / tutorial No. 6

Grade: AA / AB / BB / BC / CC / CD /DD

**Signature of the Staff In-charge with date**

**Experiment No.:6**

|  |
| --- |
| **TITLE:** IP classes and Implementation of Subnet mask concept. |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**AIM:** To study IP classes and Implementation of Subnet mask concept.

An IP (Internet Protocol) address is a unique identifier for a node or host connection on an IP network. Subnetting an IP Network can be done for a variety of reasons, including organization, use of different physical media (such as Ethernet, FDDI, WAN, etc.), preservation of address space, and security. The most common reason is to control network traffic. In an Ethernet network, all nodes on a segment see all the packets transmitted by all the other nodes on that segment. Performance can be adversely affected under heavy traffic loads, due to collisions and the resulting retransmissions. A router is used to connect IP networks to minimize the amount of traffic each segment must receive.

This experiment enables student for identifying the class of the IP address and design particular subnets as per user requirements.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Expected Outcome of Experiment:**

**CO: Demonstrate various network layer protocols and network design using IP addressing concepts.**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Books/ Journals/ Websites referred:**

1. A. S. Tanenbaum, “Computer Networks”, Pearson Education, Fourth Edition
2. B. A. Forouzan, “Data Communications and Networking”, TMH, Fourth Edition

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Pre Lab/ Prior Concepts:** IP Address, Classes, Subnet concept

**New Concepts to be learned:** Subnet mask calculation, Subnet address calculation

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Stepwise-Procedure:**

Applying a subnet mask to an IP address allows to identify the network and node parts of the address. The network bits are represented by the 1s in the mask, and the node bits are represented by the 0s. Performing a bitwise logical AND operation between the IP address and the subnet mask results in the *Network Address* or Number.

Default subnet masks:

**Class A** - 255.0.0.0 - 11111111.00000000.00000000.00000000

**Class B** - 255.255.0.0 - 11111111.11111111.00000000.00000000

**Class C** - 255.255.255.0 - 11111111.11111111.11111111.00000000

Additional bits can be added to the default subnet mask for a given Class to further subnet, or break down, a network. When a bitwise logical AND operation is performed between the subnet mask and IP address, the result defines the *Subnet* *Address* (also called the *Network Address* or *Network Number*). There are somerestrictions on the subnet address. Node addresses of all "0"s and all "1"s are reserved for specifying the local network (when a host does not know its network address) and all hosts on the network (broadcast address), respectively. This also applies to subnets. A subnet address cannot be all "0"s or all "1"s. This also implies that a 1 bit subnet mask is not allowed. This restriction is required because older standards enforced this restriction. Recent standards that allow use of these subnets have superseded these standards, but many "legacy" devices do not support the newer standards. If you are operating in a controlled environment, such as a lab, you can safely use these restricted subnets.

**CIDR** -- **C**lassless **I**nter **D**omain **R**outing:

The "classful" system of allocating IP addresses can be very wasteful; Under supernetting, the classful subnet masks are extended so that a network address and subnet mask could, for example, specify multiple Class C subnets with one address.

For example, If about 1000 addresses are required, it could be possible to supernet 4 Class C networks together:

192.60.128.0 (11000000.00111100.10000000.00000000)Class C subnet address 192.60.129.0(11000000.00111100.10000001.00000000) Class C subnet address 192.60.130.0(11000000.00111100.10000010.00000000) Class C subnet address 192.60.131.0(11000000.00111100.10000011.00000000) Class C subnet address

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

192.60.128.0 (11000000.00111100.10000000.00000000) Supernetted subnet address

255.255.252.0 (11111111.11111111.11111100.00000000)Subnet Mask 192.60.131.255 (11000000.00111100.10000011.11111111) Broadcast address

In this example, the subnet 192.60.128.0 includes all the addresses from 192.60.128.0 to 192.60.131.255. In the binary representation of the subnet mask, the Network portion of the address is 22 bits long, and the host portion is 10 bits long. Under CIDR, the subnet mask notation is reduced to simplified shorthand. Instead of spelling out the bits of the subnet mask, it is simply listed as the number of 1s bits that start the mask. In the above example, instead of writing the address and subnet mask as 192.60.128.0, Subnet Mask 255.255.252.0 .the network address would be written simply as: 192.60.128.0/22 Which indicates starting address of the network, and number of 1s bits (22) in the network portion of the address. Subnet mask in binary

11111111.11111111.11111100.00000000.

The use of a CIDR notated address is the same as for a Classful address. Classful addresses can easily be written in CIDR notation as Class A = /8, Class B = /16, and Class C = /24

To calculate the number of subnets or nodes,

No. of Nodes/ Subnets =2n-2

Where n = number of bits in either field.

Multiplying the number of subnets by the number of nodes available per subnet gives you the total number of nodes available for your class and subnet mask. Also, note that although subnet masks with non-contiguous mask bits are allowed, they are not recommended.

|  |  |  |
| --- | --- | --- |
| Example: |  |  |
| 10001100.10110011.11011100.11001000 | 140.179.220.200 | IP Address |
| 11111111.11111111.**111**00000.00000000 | 255.255.**224**.000 | Subnet Mask |
| 10001100.10110011.11000000.00000000 | 140.179.192.000 | Subnet Address |
| 10001100.10110011.11011111.11111111 | 40.179.223.255 | Broadcast Address |
|  |  |  |

1. Program starts with taking IP address from user and the number of subnets from the user.
2. Then the calculation for subnet mask is done as specified in methodology.
3. Then with AND ing with subnet mask the subnet addresses are calculated.

**IMPLEMENTATION:** (printout of code)

*import* math

*# Pargat Singh Dhanjal*

address\_cidr = input("Enter IP address(CIDR format): ")

a = address\_cidr.split('/')

ip\_address = a[0]

x = int(a[1])

flag = 0

count = 0

unused = 0

unused = unused+pow(2, 32-x)

ipLower = ""

ipHigher = ""

lower = []

higher = []

*if* len(ip\_address) > 15:

flag = 1

*for* i *in* ip\_address:

*if* i == ".":

count = count + 1

*if* count != 3:

flag = 1

octet = ip\_address.split(".", 4)

*for* i *in* range(0, len(octet)):

octet[i] = int(octet[i])

*if* octet[i] > 255:

flag = 1

created\_mask = []

tmp = []

*if* flag == 0:

print("Classless addressing")

lower = []

higher = []

ipLower = ""

ipHigher = ""

*for* i *in* range(0, x):

tmp.append("1")

*for* i *in* range(x, 32):

tmp.append("0")

print("IP address in binary form:")

*for* i *in* octet:

print(format(int(i), '08b'), end="")

print("\nSubnet address in binary:")

*for* i *in* tmp:

print(i, end="")

result = []

*for* i *in* range(0, len(tmp), 8):

result.append("".join(tmp[i:i+8]))

*for* i *in* range(0, len(result)):

*# int(num,2) for binary to decimal*

created\_mask.append(int(result[i], 2))

print("\nComplement of Subnet address in binary:")

*for* i *in* range(0, len(created\_mask)):

print(format(int(255 - created\_mask[i]), '08b'), end="")

*for* i *in* range(0, len(octet)):

*# and operation for 1st address*

lower.append(str(octet[i] & created\_mask[i]))

ipLower = ipLower + lower[0] + "." + \

lower[1] + "." + lower[2] + "." + lower[3]

print("\nFirst address: " + ipLower)

*for* i *in* range(0, len(octet)):

*# or operation with ip address and complement of subnet*

higher.append(str(octet[i] | (255 - created\_mask[i])))

ipHigher = ipHigher + higher[0] + "." + \

higher[1] + "." + higher[2] + "." + higher[3]

print("Last address: " + ipHigher)

print()

n = int(input("\nEnter no. of sub blocks: "))

subblock = [int(i) *for* i *in* lower]

*for* \_ *in* range(n):

print("\n\nTotal available IP:", unused)

number = int(input("\nEnter no. of IP address needed: "))

b = 0

x = 0

*while* 2 \*\* x < number:

x = x + 1

*if* pow(2, x) <= unused:

unused -= pow(2, x)

ipLower = str(subblock[0]) + "." + str(

subblock[1]) + "." + str(subblock[2]) + "." + str(subblock[3])

m = pow(2, x)

*while* (m):

*if* (m == 1):

ipHigher = str(subblock[0]) + "." + str(

subblock[1]) + "." + str(subblock[2]) + "." + str(subblock[3])

subblock[3] = subblock[3] + 1

*if* subblock[3] > 255:

subblock[3] = 0

subblock[2] = subblock[2] + 1

*if* subblock[2] > 255:

subblock[2] = 0

subblock[1] = subblock[1] + 1

*if* subblock[1] > 255:

subblock[2] = 0

subblock[1] = subblock[1] + 1

m -= 1

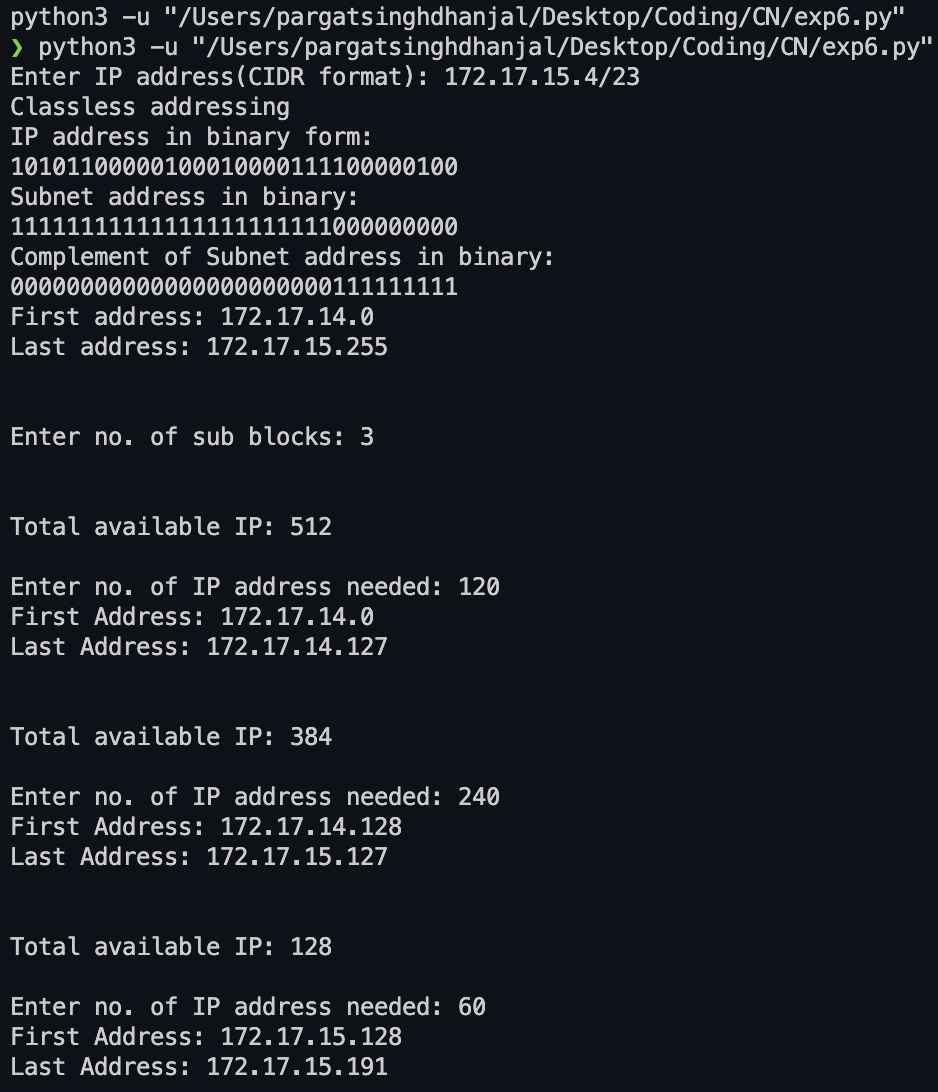
print("First Address:", ipLower)

print("Last Address:", ipHigher)

*else*:

print("Required no of addresses not available!")

Output:



**CONCLUSION:** Successfully studied IP classes and implemented Subnet mask concept.

**Post Lab Questions**

1. Which of the following is private IP address?

A. 12.0.0.1 B. 168.172.19.39

C. 172.15.14.36  **D. 192.168.24.43**

1. Which class of IP address provides a maximum of only 254 host addresses per network ID?

A. Class A

B. Class B

**C. Class C**

D. Class D

1. What is the address range of a Class B network address in binary?

A. 01xxxxxx

B. 0xxxxxxx

**C. 10xxxxxx**

D. 110xxxxx

1. Which two statements describe the IP address 10.16.3.65/23?

1.The subnet address is 10.16.3.0 255.255.254.0.

2.The lowest host address in the subnet is 10.16.2.1 255.255.254.0.

3.The last valid host address in the subnet is 10.16.2.254 255.255.254.0.

4.The broadcast address of the subnet is 10.16.3.255 255.255.254.0.

A. 1 and 3

B. 2 and 4

**C. 1, 2 and 4**

D. 2, 3 and 4

1. What is the maximum number of IP addresses that can be assigned to hosts on a local subnet that uses the 255.255.255.224 subnet mask?

A. 14 B. 15

C. 16 **D. 30**

1. You need to subnet a network that has 5 subnets, each with at least 16 hosts. Which classful subnet mask would you use?

**A. 255.255.255.192** B. 255.255.255.224

C. 255.255.255.240 D. 255.255.255.248

1. You have a network that needs 29 subnets while maximizing the number of host addresses available on each subnet. How many bits must you borrow from the host field to provide the correct subnet mask?

A. 2 B. 3

C. 4 **D. 5**

**Date: \_\_\_\_\_\_\_\_\_\_\_\_\_ Signature of faculty in-charge**

Batch: A2 Roll No.: 16010121045

Experiment / assignment / tutorial No 7

Grade: AA / AB / BB / BC / CC / CD /DD

**Signature of the Staff In-charge with date**

**Experiment No.:7**

|  |
| --- |
| **TITLE:** Study Cisco Switch Router Configuration Command using Cisco packet tracer |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**AIM:** To study basic Cisco Switch & Router configuration Commands and configure

1. Virtual LAN (VLAN).
2. Static Routing

**Expected Outcome of Experiment:**

**CO:**

**Books/ Journals/ Websites referred:**

1. S. Tanenbaum, “Computer Networks”, Pearson Education, Fourth Edition
2. Forouzan, “Data Communications and Networking”, TMH, Fourth Edition

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Pre Lab/ Prior Concepts:**  Basics of Routing and Cisco Packet Tracer

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**New Concepts to be learned:** Different Modes of Operation of Cisco router

**Cisco IOS Modes of Operation:**

* The Cisco IOS software provides access to several different command modes. Each command mode provides a different group of related commands.
* For security purposes, the Cisco IOS software provides two levels of access to commands:
  + User mode
  + Privileged mode
* The unprivileged user mode is called user EXEC mode. The privileged mode is called privileged EXEC mode and requires a password. The commands available in user EXEC mode are a subset of the commands available in privileged EXEC mode.
* The following table describes some of the most commonly used modes, how to enter the modes, and the resulting prompts. The prompt helps you identify which mode you are in and, therefore, which commands are available to you

|  |  |  |  |
| --- | --- | --- | --- |
| **Modes of Operation** | **Usage** | **How to enter the mode** | **Prompt** |
| **User EXEC** | Change terminal settings on a temporary basis, perform basic tests, and list system information. | First level accessed. | Router> |
| **Privileged EXEC** | System administration, set operating parameters. | From user EXEC mode, enter enable password command | Router# |
| **Global Config** | Modify configuration that affect the system as a whole. | From privileged EXEC, enter configure terminal. | Router(config)# |
| **Interface Config** | Modify the operation of an interface. | From global mode, enter interface type number. | Router(config-if)# |
| **Setup** | Create the initial configuration. | From privileged EXEC mode, enter command setup. | Prompted dialog |

**User EXEC Mode:**

When you are connected to the router, you are started in user EXEC mode. The user EXEC commands are a subset of the privileged EXEC commands.

**Privileged EXEC Mode:**

Privileged commands include the following:

• Configure – Changes the software configuration.

• Debug – Display process and hardware event messages.

• Setup – Enter configuration information at the prompts.

Enter the command disable to exit from the privileged EXEC mode and return to user EXEC mode.

**Configuration Mode:**

Configuration mode has a set of sub-modes that you use for modifying interface settings, routing protocol settings, line settings, and so forth. Use caution with configuration mode because all changes you enter take effect immediately.

To enter configuration mode, enter the command configure terminal and exit by pressing Ctrl-Z.

**Note:** Almost every configuration command also has a no form. In general, use the no form to disable a feature or function. Use the command without the keyword no to re-enable a disabled feature or to enable a feature that is disabled by default. For example, IP routing is enabled by default. To disable IP routing, enter the no IP routing command and enter IP routing to re-enable it.

1. **Virtual LAN (VLAN):**

A virtual local area network (VLAN) is a LAN which is not configured by physical wiring but it is configured by software. A VLAN is logical group of network devices that appear to be on same LAN despite their geographical distribution. A VLAN is implemented so that network administrators can connect a group of host in the same domain inspite of their physical location to achieve scalability and improve security features.

To subdivide a network into virtual LANs, one configures a network switch or router. Simpler network devices can partition only per physical port (if at all) , in which case each VLAN is connected with a dedicated network cable ( and VLAN connectivity is limited by the number of hardware ports available) More sophisticated devices can mark packets through tagging, so that a single interconnect ( trunk) may be used to transport data for multiple VLANs. VLAN can greatly simplify network design and deployment, because VLAN membership can be configured through software.

**Stepwise-Procedure:**

1. **Creating a simple LAN network using packet tracer:**

**Step 1:** Select 12 PCs from the end devices and one fast ethernet switch (2950/24 ports)

**Step 2:** Connect PCs and switch via copper cable from the panel. Connection can be verified by appearance of all green dots on the links.

**Step 3:** For PCs to communicate click on PC0.

* Dialog box for PC0 appears.
* Click on desktop applications by packet tracer.
* Go to IP configuration.
* Enter IP address to identify host i.e., PC0 (for example: 192.168.1.1)
* Subnet mask-by default already set one can change it as per his/her specification.

**Step 4:** Repeat step 3 for PC1

**Step 5:** Ping the PCs and check their working status.

**Step 6:** Simple PDU (Protocol Data Unit) to simulate network traffic by sending ICMP PDU to assess the network traffic. View simulation in simulation mode

**Step 7:** Configure two VLAN in a switch in 6 verticals.

**Step 8:** As per design, assign membership of VLAN to port using following command.

# switch port access vlan2 or vlan3

**Step 9:** Check the status of VLAN.

1. **Static Routing Configuration**

**IMPLEMENTATION:** (printout of code)

**A computer screen with white text

Description automatically generatedA close-up of a computer

Description automatically generatedA group of computers on a desk

Description automatically generatedA computer screen with white text

Description automatically generatedA computer screen with white text

Description automatically generatedA computer screen with text on it

Description automatically generatedA computer screen shot of a computer

Description automatically generatedA computer screen shot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA computer screen shot of a black screen

Description automatically generatedA computer screen shot of a computer

Description automatically generated**

**CONCLUSION:**

With the help of this experiment we were able to learn, understand and implement the following details

* basic Cisco Router configuration Commands
* Implemented Static Routing using Cisco Packet Tracer

**Date: \_\_\_\_\_\_\_\_\_\_\_\_\_ Signature of faculty in-charge**

Batch: A2 Roll No.: 16010121045

Experiment / assignment / tutorial No. 8

Grade: AA / AB / BB / BC / CC / CD /DD

**Signature of the Staff In-charge with date**

**Experiment No.:8**

|  |
| --- |
| **TITLE: Study and configure RIP protocol using Cisco Packet tracer** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_AIM:** To study and configure RIP protocol using Cisco Packet tracer

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Expected Outcome of Experiment:**

**CO: To demontsrate understanding of static routing using cisco packet tracer**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Books/ Journals/ Websites referred:**

1. A. S. Tanenbaum, “Computer Networks”, Pearson Education, Fourth Edition
2. B. A. Forouzan, “Data Communications and Networking”, TMH, Fourth Edition

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Pre Lab/ Prior Concepts:**

IPv4 Addressing, Subnetting, Distance Vector Protocol, Router configuration Commands.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**New Concepts to be learned:** RIP Protocol and its configuration.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**RIP (Routing Information Protocol)**

RIP is a standardized Distance Vector protocol, designed for use on smaller networks. RIP was one of the first true Distance Vector routing protocols and is supported on a wide variety of systems.

RIP adheres to the following Distance Vector characteristics:

• RIP sends out periodic routing updates (every 30 seconds)

• RIP sends out the full routing table every periodic update.

• RIP uses a form of distance as its metric (in this case, hop count).

• RIP uses the Bellman-Ford Distance Vector algorithm to determine the best “path” to a particular destination

Other characteristics of RIP include:

• RIP supports IP and IPX routing.

• RIP utilizes UDP port 520

• RIP routes have an administrative distance of 120.

• RIP has a maximum hop count of 15 hops.

**RIP Versions**

RIP has two versions, Version 1 (RIPv1) and Version 2 (RIPv2).

RIPv1 (RFC 1058) is ***classful***, and thus does not include the subnet mask with its routing table updates. Because of this, RIPv1 does not support **Variable Length Subnet Masks (VLSMs)**. When using RIPv1, networks must be contiguous, and subnets of a major network must be configured with identical subnet masks. Otherwise, route table inconsistencies (or worse) will occur.

RIPv1 sends updates as broadcasts to address 255.255.255.255.

RIPv2 (RFC 2543) is ***classless***, and thus does include the subnet mask with its routing table updates. RIPv2 fully supports VLSMs, allowing discontiguous networks and varying subnet masks to exist.

Other enhancements offered by RIPv2 include:

• Routing updates are sent via multicast, using address 224.0.0.9

• Encrypted authentication can be configured between RIPv2 routers

• Route tagging is supported.

RIPv2 can interoperate with RIPv1. By default:

• RIPv1 routers will send only Version 1packets

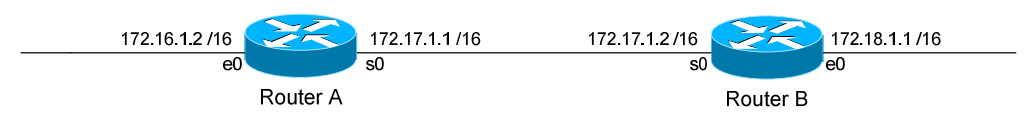
• RIPv1 routers will receive both Version 1 and 2 updates

• RIPv2 routers will both send and receive only Version 2 updates

We can control the version of RIP a particular interface will “send” or “receive.”

Unless RIPv2 is manually specified, a Cisco will default to RIPv1 when configuring RIP.

**RIPv1 Basic Configuration**



Routing protocol configuration occurs in Global Configuration mode. On Router A, to configure RIP, we would type:

**Router(config)#** router rip

**Router(config-router)#** network 172.16.0.0

**Router(config-router)#** network 172.17.0.0

The first command, router rip, enables the RIP process.

The network statements tell RIP which networks you wish to advertise to other RIP routers. We simply list the networks that are directly connected to our router. Notice that we specify the networks at their classful boundaries, and we do not specify a subnet mask.

To configure Router B:

**Router(config)#** router rip

**Router(config-router)#** network 172.17.0.0

**Router(config-router)#** network 172.18.0.0

The routing table on Router A will look like:

**RouterA#** show ip route

Gateway of last resort is not set

C 172.16.0.0 is directly connected, Ethernet0

C 172.17.0.0 is directly connected, Serial0

R 172.18.0.0 [120/1] via 172.17.1.2, 00:00:00, Serial0

The routing table on Router B will look like:

**RouterB#** show ip route

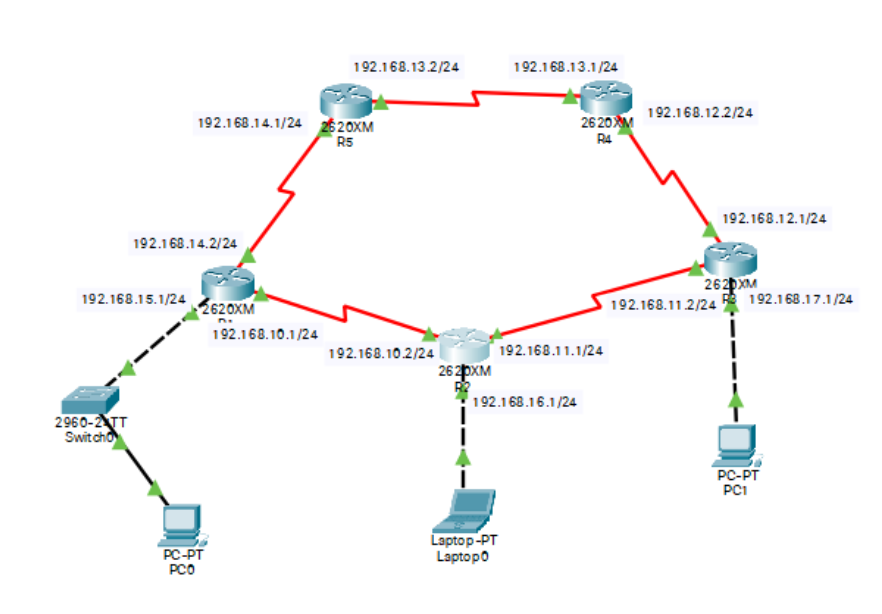
Gateway of last resort is not set

C 172.17.0.0 is directly connected, Serial0

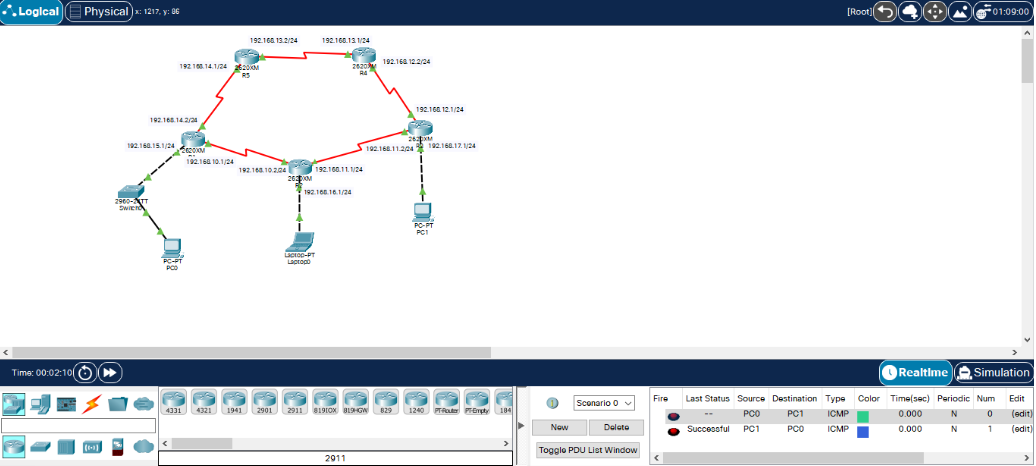
C 172.18.0.0 is directly connected, Ethernet0

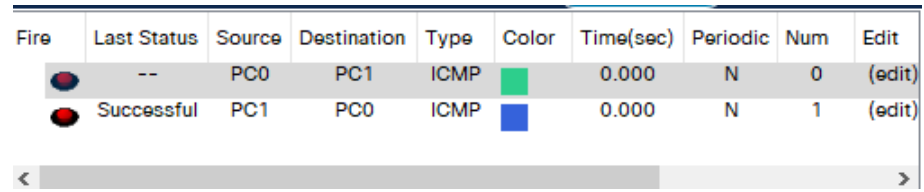
R 172.16.0.0 [120/1] via 172.17.1.1, 00:00:00, Serial0

**IMPLEMENTATION: (**printout of code)

****

**Message sent from PC1 to PC0 is successful.**

****

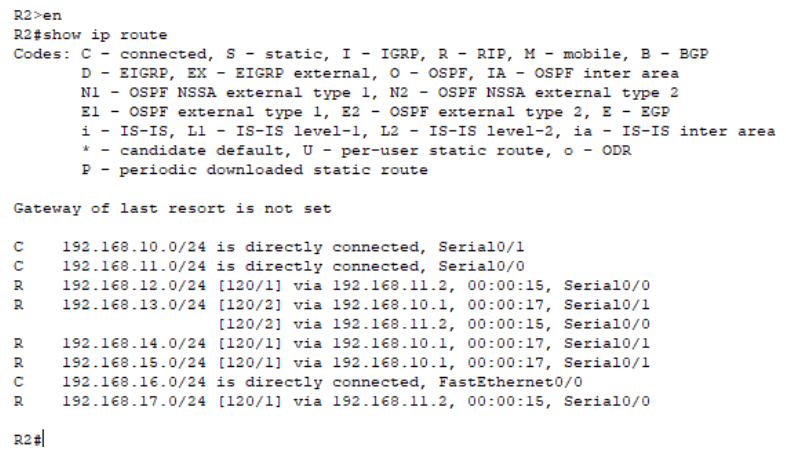
****

**Dynamic Routing table entries for each router in designed network:**

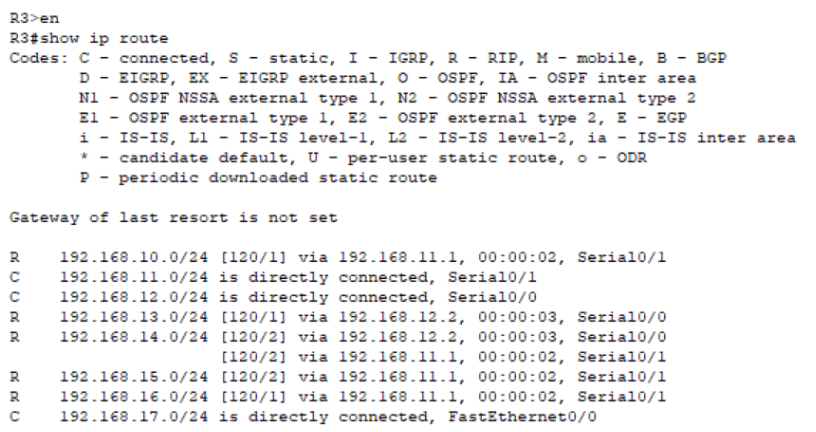
**For router R1:**

****

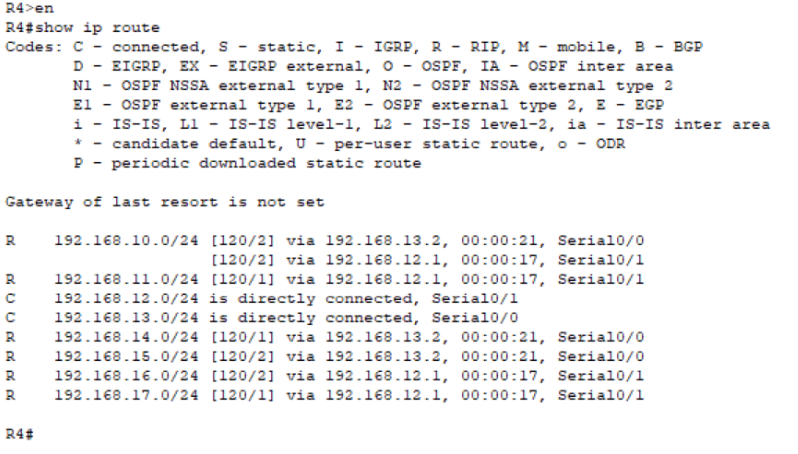
**For router R2:**

****

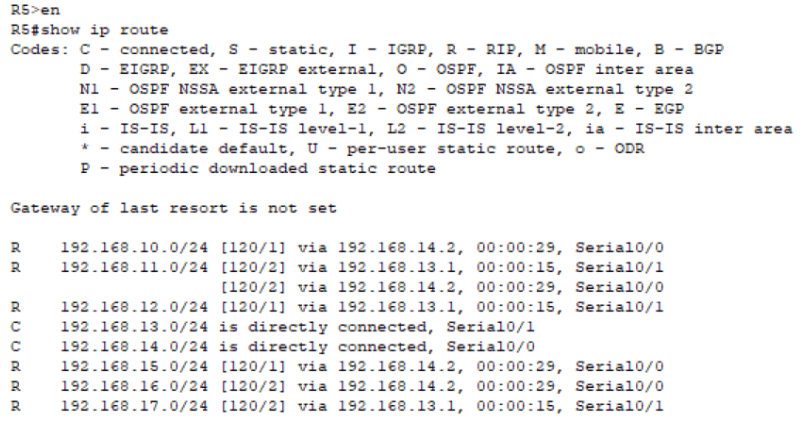
**For router R3:**

****

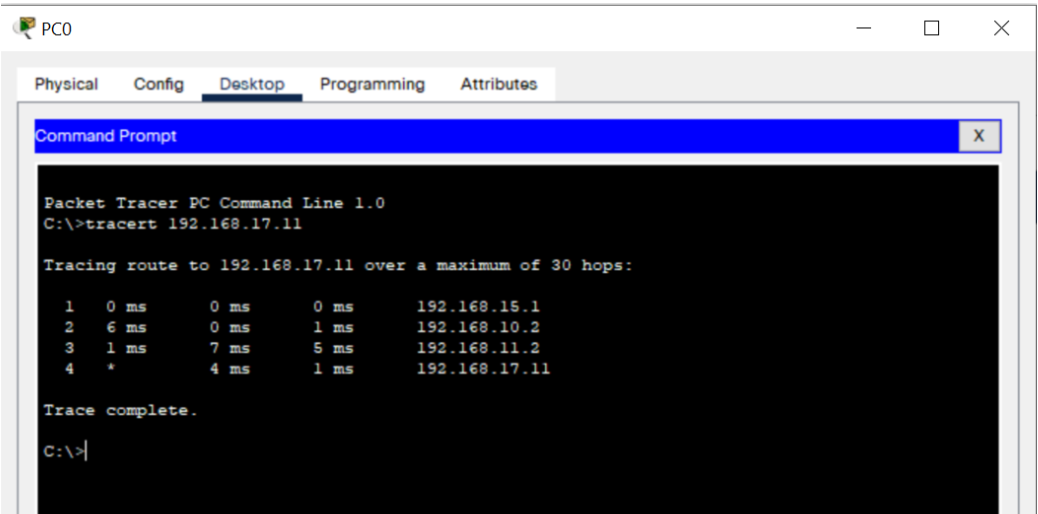
**For router R4:**

****

**For router R5:**

****

**Tracing route from PC0 to PC1:**

****

**CONCLUSION:** Successfully studied and configure RIP protocol using Cisco Packet tracer.

**Post Lab Questions**

1. ............ are two popular examples of distance vector routing protocols.  
A. OSPF and RIP  
**B. RIP and BGP**C. BGP and OSPF  
D. BGP and SPF

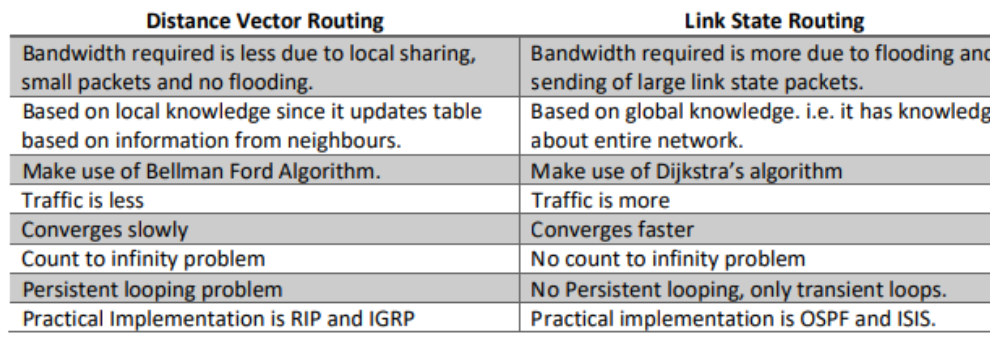
2. A ......... routing table contains information entered manually.  
**A. static**  
B. dynamic  
C. hierarchical  
D. non static

3. A .......... routing table is updated periodically using one of the dynamic routing protocols.  
A. static  
**B. dynamic**  
C. hierarchical  
D. non static

4. Which of the following is not the category of dynamic routing algorithm.  
A. Distance vector protocols  
B. Link state protocols  
C. Hybrid protocols  
**D. Automatic state protocols**

5. In .......... forwarding, the mask and destination addresses are both 0.0.0.0 in the routing table.  
A. next-hop  
B. network-specific  
C. host-specific  
**D. default**

6. Differentiate between Distance Vector Routing and Link State Routing.

****

**Date: \_\_\_\_\_\_\_\_\_\_\_ Signature of faculty in-charge**

Batch: A2 Roll No: 16010121045

Experiment / assignment / tutorial No: 9

Grade: AA / AB / BB / BC / CC / CD /DD

**Signature of the Staff In-charge with date**

**Experiment No.:9**

|  |
| --- |
| **TITLE: Study and configure DHCP & DNS protocol using Cisco Packet tracer** |

**AIM:** To study and configure **DHCP/DNS** protocol using Cisco Packet tracer

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Expected Outcome of Experiment:**

**CO:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Books/ Journals/ Websites referred:**

1. A. S. Tanenbaum, “Computer Networks”, Pearson Education, Fourth Edition
2. B. A. Forouzan, “Data Communications and Networking”, TMH, Fourth Edition

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Pre Lab/ Prior Concepts:**

IPv4 Addressing, Subnetting, Link State Protocol, Router configuration Commands

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**New Concepts to be learned: DHCP/DNS** Protocol and its configuration.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**THEORY:**

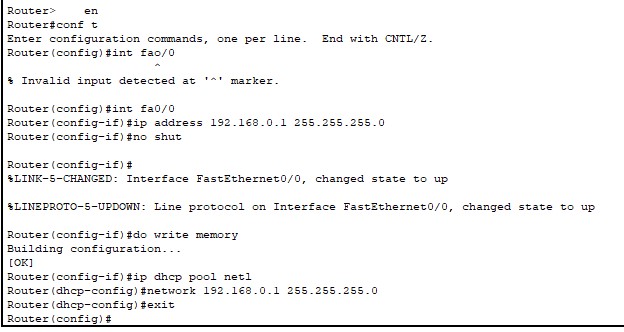
Dynamic Host Configuration Protocol (DHCP) is a client/server protocol that automatically provides an Internet Protocol (IP) host with its IP address and other related configuration information such as the subnet mask and default gateway.

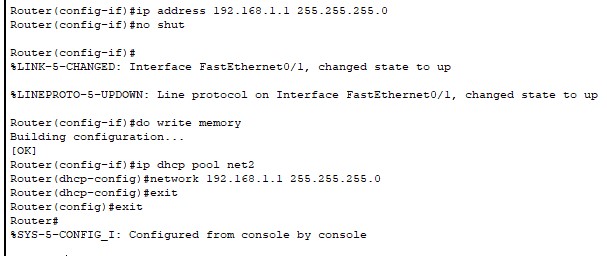
All computers on the Internet, from your smart phone or laptop to the servers that serve content for massive retail websites, find and communicate with one another by using numbers. These numbers are known as IP addresses. When you open a web browser and go to a website, you don't have to remember and enter a long number. Instead, you can enter a domain name like example.com and still end up in the right place.

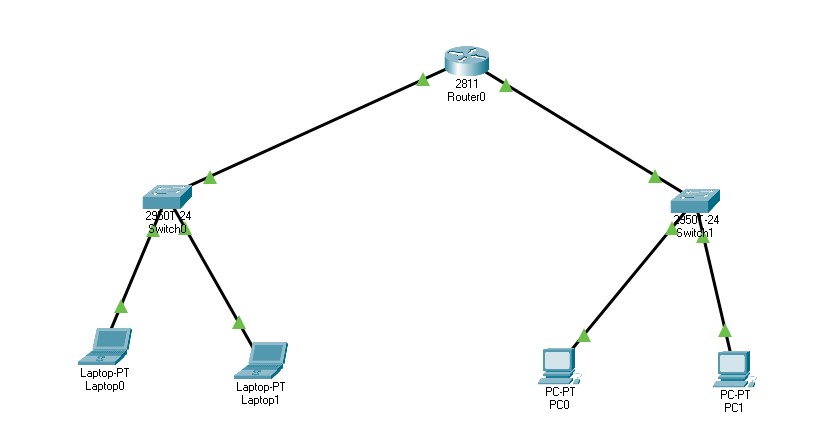
A DNS service such as Amazon Route 53 is a globally distributed service that translates human readable names like www.example.com into the numeric IP addresses like 192.0.2.1 that computers use to connect to each other. The Internet’s DNS system works much like a phone book by managing the mapping between names and numbers. DNS servers translate requests for names into IP addresses, controlling which server an end user will reach when they type a domain name into their web browser. These requests are called queries.

**IMPLEMENTATION:**

The cli for router 1 assigning the ip range for each fast Ethernet







Requesting the IP from DHCP

A screenshot of a computer

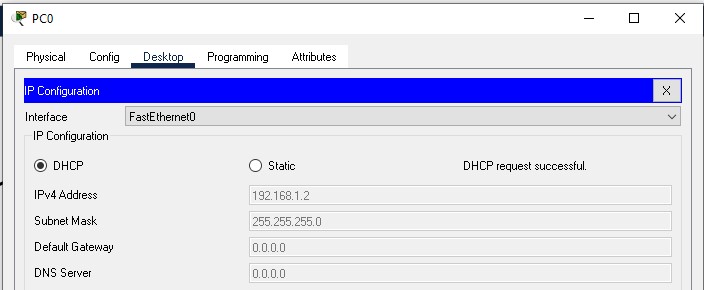
Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated



A screenshot of a computer

Description automatically generated

**CONCLUSION:**

**We successfully studied and configured DHCP/DNS protocol using Cisco Packet tracer**

**Date: \_\_\_\_\_\_\_\_\_\_\_ Signature of faculty in-charge**

Batch: A2 Roll No.: 16010121045

Experiment / assignment / tutorial No.\_\_\_\_\_\_\_

Grade: AA / AB / BB / BC / CC / CD /DD

**Signature of the Staff In-charge with date**

**Experiment No.:10**

|  |
| --- |
| **TITLE: Study of Packet Analyzer tool: Wireshark** |

**AIM:** To study and analyse various Protocols using Packet Analyzer tool: Wireshark

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Expected Outcome of Experiment:**

**CO:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Books/ Journals/ Websites referred:**

1. A. S. Tanenbaum, “Computer Networks”, Pearson Education, Fourth Edition
2. B. A. Forouzan, “Data Communications and Networking”, TMH, Fourth Edition

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Pre Lab/ Prior Concepts:**

IPv4 Addressing, Subnetting, Link State Protocol, Router configuration Commands

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**New Concepts to be learned: Packet Analyzer tool: Wireshark**.

**THEORY:**

Wireshark is an open-source packet analyzer, which is used for education, analysis, software development, communication protocol development, and network troubleshooting. It is used to track the packets so that each one is filtered to meet our specific needs. It is commonly called as a sniffer, network protocol analyzer, and network analyzer. It is also used by network security engineers to examine security problems. Wireshark is a free to use application which is used to apprehend the data back and forth. It is often called as a free packet sniffer computer application. It puts the network card into an unselective mode, i.e., to accept all the packets which it receives.

**Uses of Wireshark**

1. It is used by network security engineers to examine security problems.
2. It allows the users to watch all the traffic being passed over the network.
3. It is used by network engineers to troubleshoot network issues.
4. It also helps to troubleshoot latency issues and malicious activities on your network.
5. It can also analyze dropped packets.
6. It helps us to know how all the devices like laptop, mobile phones, desktop, switch, routers, etc., communicate in a local network or the rest of the world.

**IMPLEMENTATION:**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Using the ip header :**

Version:4

Header length:5 (20 bytes)

Different types of services: dscp ecn

Identifier:0xb150

Flags: 0x40

Fragments :reserved (not set),don’t fragment (set)

Time to live:128

Protocol:tcp

Headerchecksum:0x7c25

Source ip:172.17.14.14

Destination ip :74.125.200.188

**Tcp:**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

Source port:49924

Destination:5228

**Sequence no:**

**A close up of numbers

Description automatically generated**

**Acknowledgement:**

****

**Flags:**

**A screenshot of a computer program

Description automatically generated**

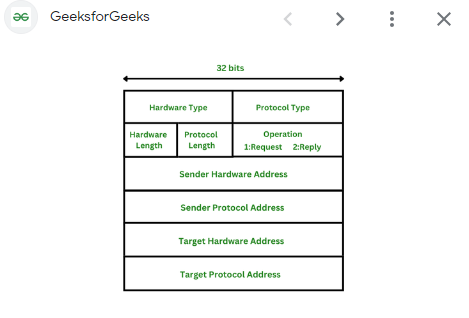
**Window:1023**

**Checksum:**

****

Urgent pointer :0

**ARP:**

****

**A screenshot of a computer

Description automatically generated**

Arp takes place in frames as the data travels where only the sender and receivers

Where the sender and recievers address is noted.

**UDP:**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**CONCLUSION:**

Thus we have implemented the wireshark experiment

**Date: \_\_\_\_\_\_\_\_\_\_\_ Signature of faculty in-charge**