RAJALAKSHMI
ENGINEERING COLLEGE
An AUTONOMOUS Institution
Affiliated to ANNA UNIVERSITY, Chennai

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

# CS19541- COMPUTER NETWORKS LABORATORY

## LAB MANUAL

THIRD YEAR

FIFTH SEMESTER

2024- 2025

ODD SEMESTER

Name : C Sakthi Maheswari

Year / Branch / Section : III- Computer Science and Design

Register No. : 221701048

Semester : V

Academic Year : 2024-2025

## List of Experiments

| List of Experiments | | |
|---|---|---|
| 1. | **Study of various Network commands used in Linux and Windows:**<br>Hands-on practice of various network commands. | [4] |
| 2. | **Study of Network cables.**<br>1. Understand different types of Network cables.<br>2. Make a cross-wired cable and straight through cable using clamping/crimping tool. | [4] |
| 3. | **Experiments on CISCO PACKET TRACER (Simulation Tool):**<br>a) To understand environment of CISCO PACKET TRACER to design simple network.<br>b) Analyse the behaviour of network devices using CISCO PACKET TRACER simulator.<br>Design a simple network with multiple nodes and connect via networking devices available in library. Perform simulation and trace communication behaviour of specified network devices.<br>1: Use only HUB to design a small network having 4 to 6 hosts<br>2: Use only a Switch to design a small network with 4 to 6 hosts.<br>3: Use both the device (HUB and SWITCH) for a network and find out functioning difference between switch and hub.<br>**Find out the network topology implemented in your college and draw and label that topology in your observation book.** | [2]<br>[2] |

| | | |
|---|---|---|
| 4. | **a) Setup and configure a LAN (Local area network) using a Switch and Ethernet cables in your lab.**<br>1. Connect 3-4 host machines to a switch.<br>2. Assign ip addresses to each host machine.<br>3. Check the connectivity between the machines by using ping command.<br>4. Share and access files and folder across the machines of the LAN. | [2] |

| | | |
|---|---|---|
| 5. | **Experiments on Packet capture tool: Wireshark**<br>To understand the features of wireshark as a packet capture tool and understand **encapsulation of information at various layers of a Protocol stack**. | [4] |
| 6. | **Error Correction at Data Link Layer:**<br>Write a program to implement error detection and correction using HAMMING code concept. Make a test run to input data stream and verify error correction feature. | [4] |
| 7. | **Flow control at Data Link Layer:**<br>Write a program to implement flow control at data link layer using SLIDING WINDOW PROTOCOL. Simulate the flow of frames from one node to another. | [4] |
| 8. | a) **Virtual LAN**:<br>Simulate Virtual LAN configuration using CISCO Packet Tracer Simulation.<br>b)<br>There are 10 faculty in Robotics department sitting in 3 different blocks. Design and configure a Virtual LAN for Robotics department (using switch and Ethernet cables) so that all the faculty are logically in the same LAN.<br>c) **Wireless LAN:**<br>Configuration of Wireless LAN using CISCO Packet Tracer. | [4]<br>[2] |

| | | |
|---|---|---|
| 9. | **Implementation of SUBNETTING in CISCO PACKET TRACER simulator.**<br>a) Design multiple subnet with suitable number of hosts.<br>b) Assign static IP address across all subnet and connect the subnets via Router.<br>c) Simulate packet transmission across the subnets and observe the results:-<br>a. When subnets are connected via a router.<br>b. When subnets are not connected without a router. | [4] |

| | | |
|---|---|---|
| 10. | **Internetworking with routers in CISCO PACKET TRACER simulator.**<br>**a) Design and configure a simple internetwork using a router.** 1. Design different networks (with 3 to 4 hosts) and connect via Router.<br>2. Allot static ip address to machines and router interfaces.<br>3. Perform simulation and trace how routing is done in packet transmission.<br>**b) Design and configure an internetwork using wireless router DHCP server and internet cloud.**<br>**c) Design and configure an inter-network in your lab using switch, router and Ethernet cables.** | [4]<br>[2] |
| 11. | **Routing at Network Layer:**<br>a) Simulate Static Routing Protocol Configuration using CISCO Packet Tracer.<br>b) Simulate RIP using CISCO Packet Tracer. | [4] |
| 12. | **End –End Communication at Transport Layer**<br>a) Implement echo client server using TCP/UDP sockets.<br>b) Implement a chat program using socket programming. | [4] |
| 13. | Implement your own ping program. | [2] |
| 14. | Write a code using RAW sockets to implement packet sniffing. | [4] |
| 15. | Analyse various types of servers using Webalizer tool. | [4] |
| | **Total** | **60 hours** |

| Additional programs for practice | | |
|---|---|---|
| 1. | **Data Link Layer (Frame Generation):** Write a program to read a stream of data from data file (Having Characters) to create BSC frames by implementing character stuffing concept and inserting control characters. The receiving program must execute on other computer and decode received bytes and write to a file. | |
| 2. | Demonstrate Configuration of Network Address Translation (NAT) and Port Address Translation (PAT) using CISCO Packet Tracer simulation. | |
| 3. | Implement a static routing protocol which also displays the routing table details after every update. | |
| 4. | Implement a dynamic routing protocol which also displays the routing table after every updates. | |
| 5. | Implement FTP server using socket programming. | |

## Practical -1

**AIM: -**

**Study of various Network commands used in Linux and Windows:**

**BASIC NETWORKING COMMANDS:**

**arp –a:-** ARP is short form of address resolution protocol, It will show the IP address of your computer along with the IP address and MAC address of your router.

**hostname:** This is the simplest of all TCP/IP commands. It simply displays the name of your computer.

**ipconfig /all:** This command displays detailed configuration information about your TCP/IP connection including Router, Gateway, DNS, DHCP, and type of Ethernet adapter in your system

**nbtstat –a:** This command helps solve problems with NetBIOS name resolution. (Nbt stands for NetBIOS over TCP/IP)

**netstat:** (network statistics) netstat displays a variety of statistics about a computers active TCP/IP connections. It is a command line tool for monitoring network connections both incoming and outgoing as well as viewing routing tables, interface statistics etc. e.g.:- netstat -r

**nslookup:** (name server lookup) is a tool used to perform DNS lookups in Linux. It is used to display DNS details, such as the IP address of a particular computer, the MX records for a domain or the NS servers of a domain. nslookup can operate in two modes: interactive and non-interactive.

e.g.:- nslookup www.google.com

**pathping:** Pathping is unique to Window's, and is basically a combination of the Ping and Tracert commands. Pathping traces the route to the destination address then launches a 25 second test of each router along the way, gathering statistics on the rate of data loss along each hop.

**ping:** (Packet INternet Groper) command is the best way to test connectivity between two
nodes. Ping use ICMP (Internet Control Message Protocol) to communicate to other devices.

1. #ping hostname( ping localhost)
2. #ping ip address (ping 4.2.2.2)
3. #ping fully qualified domain name(ping www.facebook.com

**Route:** route command is used to show/manipulate the IP routing table. It is primarily used to setup static routes to specific host or networks via an interface.

## Some important Linux networking commands

### 1. ip

The ip command is one of the basic commands every administrator will need in daily work, from setting up new systems and assigning IPs to troubleshooting existing systems. The ip command can show address information, manipulate routing, plus display network various devices, interfaces, and tunnels.

**ip <OPTIONS> <OBJECT> <COMMAND>**

Here are some common use cases for the ip command.

a. To show the IP addresses assigned to an interface on your server:
   a. [root@server ~]# *ip address show*

b. To assign an IP to an interface, for example, **enps03**:
   a. [root@server ~]# ip address add 192.168.1.254/24 dev enps03

c. To delete an IP on an interface:
   a. [root@server ~]# ip address del 192.168.1.254/24 dev enps03

d. Alter the status of the interface by bringing the interface **eth0** online: [root@server ~]# ip link set eth0 up

e. Alter the status of the interface by bringing the interface **eth0** offline: [root@server ~]# ip link set eth0 down

f. Alter the status of the interface by enabling promiscuous mode for **eth0**: [root@server ~]# ip link set eth0 promisc on

g. Add a default route (for all addresses) via the local gateway 192.168.1.254 that can be reached on device **eth0**:
   [root@server ~]# ip route add default via 192.168.1.254 dev eth0

h. Add a route to 192.168.1.0/24 via the gateway at 192.168.1.254:
   [root@server ~]# ip route add 192.168.1.0/24 via 192.168.1.254

i. Add a route to 192.168.1.0/24 that can be reached on device **eth0**: [root@server ~]# ip route add 192.168.1.0/24 dev eth0

j. Delete the route for 192.168.1.0/24 via the gateway at 192.168.1.254:

```
[root@server ~]# ip route delete 192.168.1.0/24 via 192.168.1.254
```

k. Display the route taken for IP 10.10.1.4:

```
[root@server ~]# ip route get 10.10.1.4
```

## 2. ifconfig

The ifconfig command was/is a staple in many sysadmin's tool belt for configuring and troubleshooting networks. It has since been replaced by the ip command discussed above.

### 3. mtr

MTR (Matt's traceroute) is a program with a command-line interface that serves as a network diagnostic and troubleshooting tool. This command combines the functionality of the ping and traceroute commands. Just like a traceroute, the mtr command will show the route from a computer to a specified host. mtr provides a lot of statistics about each hop, such as response time and percentage. With the mtr command, you will get more information about the route and be able to see problematic devices along the way. If you see a sudden increase in response time or packet loss, then obviously, there is a bad link somewhere.
The syntax of the command is as follows: **mtr <options> hostname/IP**
Let's look at some common use cases.

a. The basic mtr command shows you the statistics, including each hop (hostnames) with time and loss%:

```
[root@server ~]# mtr google.com
```

b. Show numeric IP addresses (if you use -g, you will get IP addresses (numbers) instead of hostnames):

```
[root@server ~]# mtr -g google.com
```

c. Show the numeric IP addresses and hostnames, too:

```
[root@server ~]# mtr -b google.com
```

d. Set the number of pings that you want to send:

```
[root@server ~]# mtr -c 10 google.com
```

### 4. tcpdump

The tcpdump command is designed for capturing and displaying packets.

You can install tcpdump with the command below: [root@server ~]
*# dnf install -y tcpdump*

Before starting any capture, you need to know which interfaces tcpdump can use. You will need to use sudo or have root access in this case.

```
[root@server ~]# tcpdump -D
```

If you want to capture traffic on **eth0**, you can initiate that with tcpdump -i eth0 sample output:

```
[root@server ~]# tcpdump -i eth0
```

[root@server ~]# *tcpdump -i eth0*
*-c 10*

**Capture traffic to**ning from a specific host. For example, to find
**host** traffic coming

from and going to 8.8.8.8, use th*0 host 8.8.8.8*
command:
[root@server ~]# *tcpdump -i*
*eth0 -*

For traffic coming from 8.8.8.8,
use: using the command below:
[root@server ~]# *tcpdump -i eth0* *t 10.1.0.0 mask 255.255.255.0*
*host 8.8.8.8*

*t 10.1.0.0/24*
For outbound traffic going to 8.8.**nd from port numbers**
use:
[root@server ~]# *tcpdump -i eth0*
*host 8.8.8.8*

**Capture traffic t**
**netwo***st 8.8.8.8 and port 53*
You can also capture traffic to ar
network
[root@server ~]# *tcpdump* *0 host www.google.com and port 443*
or:
[root@server ~]# *tcpdump -*
**Capture traffic** *rt not 53 and not 25*
Capture only DNS port 53
traffic:
[root@server ~]# *tcpdump -i eth0*
*port 53*

For a specific host,
[root@server ~]# *tcpdump -*

To capture only HTTPS traffic,
[root@server ~]# *tcpdump -i*
*eth0 -*

To capture all port except port 80
and 25,
[root@server ~]# *tcpdump -*

### 5. ping

Ping is a tool that 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 is displayed, along with round-trip times. Ping is the primary TCP/IP command used to troubleshoot connectivity reachability, and name resolution.

```
[root@server ~]# ping google.com
PING google.com (216.58.206.174) 56(84) bytes of data.
64 bytes from sof02s27-in-f14.1e100.net (216.58.206.174): icmp_seq=1
ttl=56 time=10.7 ms
64 bytes from sof02s27-in-f14.1e100.net (216.58.206.174): icmp_seq=2
ttl=56 time=10.2 ms
64 bytes from sof02s27-in-f14.1e100.net (216.58.206.174): icmp_seq=3
 ttl=56 time=10.4 ms ^C
```

You need to stop the ping command by pressing **CTRL+C**.
Otherwise, it will ping until you stop it.

If you want to ping a host ten times, use the following command:

```
[root@server ~]# ping -c 10 google.com
```

While pinging a host, you'll find different output from the ping results, including the following three examples.

**Destination Host Unreachable**

The possible best reason is there is no route from the local host system and to the destination desired destination host, or a remote router reports that it has no route to the destination host.

**Request timed out**

This result means that no Echo Reply messages were received within the default time of one second or the time that you set while you are pinging that host. This can be due to many different causes; the most common include network congestion, failure of the ARP request, packet filtering/firewall, etc.

**Unknown host/Ping Request Could Not Find Host**

Maybe you misspelled the hostname or the host does not exist at all in the network.

You must have 0% packet loss for every ping result with a good latency or lower response time. Depending on which transmission medium (UTP, fibre optics cable, WiFi) you're using, your latency will differ.

### Configuring an Ethernet connection by using nmcli

If you connect a host to the network over Ethernet, you can manage the connection's settings on the command line by using the nmcli utility.

**Procedure**

1. List the NetworkManager connection profiles:

   ```
   # nmcli connection show
   NAME                          UUID                                TYPE
   DEVICE
   Wired connection 1    a5eb6490-cc20-3668-81f8-0314a27f3f75   ethernet
   enp1s0
   ```

2. # **nmcli connection add con-name <connection-name> ifname <device-name> type ethernet**
   Skip this step to modify an existing profile.

3. Optional: Rename the connection profile:

   # **nmcli connection modify "Wired connection 1"**
   Here, "Wired connection 1" is the name of the
   connection

4. Display the current settings of the connection profile: # **nmcli connection show**

   ```
     connection.interface-name:     enp1s0
     connection.autoconnect:        yes
     ipv4.method:                   auto
     ipv6.method:                   auto
   ...
   ```

5. Configure the IPv4 settings:
   - To use DHCP, enter:

     # **nmcli connection modify "Wired connection 1" ipv4.method auto** Skip this step if ipv4.method is already set to auto (default).

   - To set a static IPv4 address, network mask, default gateway, DNS servers, and search domain, enter:

     # **nmcli connection modify "Wired connection 1" ipv4.method manual ipv4.addresses 192.0.2.1/24 ipv4.gateway 192.0.2.254 ipv4.dns 192.0.2.200 ipv4.dns-search example.com**

6. Configure the IPv6 settings:
   - To use stateless address autoconfiguration (SLAAC), enter:

# **nmcli connection modify “Wired connection 1” ipv6.method auto** Skip this step if ipv6.method is already set to auto (default).

- To set a static IPv6 address, network mask, default gateway, DNS servers, and search domain, enter:

```
# nmcli connection modify “Wired connection 1” ipv6.method manual
ipv6.addresses 2001:db8:1::fffe/64 ipv6.gateway 2001:db8:1::fffe
ipv6.dns
2001:db8:1::ffbb ipv6.dns-search example.com
```

7. Activate the profile:

```
# nmcli connection up Internal-LAN
```

**Verification**

1. Display the IP settings of the NIC:

```
# ip address show enp1s0 enp1s0:
<BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state
UP group default qlen 1000          link/ether 52:54:00:17:b8:b6 brd ff:ff:ff:ff:ff:ff
         inet 192.0.2.1/24 brd 192.0.2.255 scope global
        noprefixroute enp1s0 valid_lft forever preferred_lft forever
        inet6 2001:db8:1::fffe/64 scope global noprefixroute
        valid_lft forever preferred_lft forever
```

2. Display the IPv4 default gateway:

```
# ip route show default

  default via 192.0.2.254 dev enp1s0 proto static metric 102
```

3. Display the IPv6 default gateway:

```
# ip -6 route show default   default via 2001:db8:1::ffee dev
enp1s0 proto static metric 102 pref medium
```

4. Display the DNS settings:

```
# cat /etc/resolv.conf
  search example.com
  nameserver 192.0.2.200
  nameserver 2001:db8:1::ffbb
```

If multiple connection profiles are active at the same time, the order of nameserver entries depend on the DNS priority values in these profile and the connection types.

5. Use the ping utility to verify that this host can send packets to other hosts:

```
# ping <host-name-or-IP-address>
```

**Troubleshooting**

- Verify that the network cable is plugged-in to the host and a switch.
- Check whether the link failure exists only on this host or also on other hosts connected to the same switch.
- Verify that the network cable and the network interface are working as expected.

  Perform hardware diagnosis steps and replace defect cables and network interface cards.
- If the configuration on the disk does not match the configuration on the device, starting or restarting NetworkManager creates an in-memory connection that reflects the configuration of the device. **Student Observation:**

1. **Which command is used to find the reachability of a host machine from your device?**
2. **Which command will be give the details of hops taken by a packet to reach its destination?**
3. **Which commands displays the ip configuration of your machine.**
4. **Which command displays the TCP port status in your machine?**
5. **Write the modify the ip configuration in a Linux machine.**

**Answer:**

1. ping
2. traceroute (or tracert on Windows)
3. ifconfig (or ip a on modern Linux systems), ipconfig (on Windows)
4. netstat -t
5. ip addr add (to modify IP configuration)

**Result:**

The experiment was studied and experimented successfully.