



**MGM's**

**Jawaharlal Nehru Engineering College  
Aurangabad**

**MGM University, Aurangabad**

**Department of Computer Science & Engineering**

**LAB MANUAL**

---

Program (UG/PG): UG

Year : Third Year

Semester : V

Course Code : 20UCS510L

Course Title : Computer Network Lab

Prepared By : Ms. A. B. Shahin (Husain)

---

Department of Computer Science & Engineering

**2022-23**

## **FOREWORD**

It is my great pleasure to present this laboratory manual for Third year engineering students for the subject of Computer networks.

As a student, many of you may be wondering with some of the questions in your mind regarding the subject and exactly what has been tried is to answer through this manual.

As you may be aware that MGM has already been awarded with ISO 9001:2000 certifications and it is our duty to technically equip our students taking the advantage of the procedural aspects of ISO 9001:2000 Certification.

Faculty members are also advised that covering these aspects in the initial stage itself will greatly relieve them in future as much of the load will be taken care of by the enthusiasm energies of the students once they are conceptually clear.

Dr. H. H. Shinde  
Principal

## **LABORATORY MANUAL CONTENTS**

This manual is intended for the Third year students of Computer Science and Engineering in the subject of Computer Network. This manual typically contains practical/Lab Sessions related Computer Network covering various aspects related to the subject to enhanced understanding.

Students are advised to thoroughly go through this manual rather than only topics mentioned in the syllabus as practical aspects are the key to understanding and conceptual visualization of theoretical aspects covered in the books.

Good Luck for your Enjoyable Laboratory Sessions

Prof. V. B. Musande  
HOD, CSE

Mrs. Asma B. Shahin(Husain)  
Asst. Prof., CSE Dept.



**Jawaharlal Nehru Engineering College, Aurangabad**  
**Department of Computer Science and Engineering**

---

**Vision of CSE Department:**

To develop computer engineers with necessary analytical ability and human values who can creatively design, implement a wide spectrum of computer systems for welfare of the society.

**Mission of the CSE Department:**

- I. Preparing graduates to work on multidisciplinary platforms associated with their professional position both independently and in a team environment.
- II. Preparing graduates for higher education and research in Computer Science and Engineering enabling them to develop systems for society development.

**Programme Educational Objectives:**

**Graduates will be able to**

- I. To analyze, design and provide optimal solution for Computer Science and Engineering and multidisciplinary problems.
- II. To pursue higher studies and research by applying knowledge of mathematics and fundamentals of computer science.
- II. To exhibit professionalism, communication skills and adapt to current trends by engaging in lifelong learning.

**Programme Outcomes (POs): Engineering Graduates will be able to:**

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

## **LIST OF EXPERIMENTS**

**Course Code: 20UCS510L**

**Course Title: Computer Networks Lab**

<b>Sr. No</b>	<b>Name of the experiment</b>	<b>Page No.</b>
1	Configure of network - Assigning IP Address, Subnet Mask, Gateway & Testing Basic Connectivity.	
2	Study of Networking Devices.	
3	Study transmission media.	
4	Creating a networking cable using crimping tool & study different connectors	
5	Use basic networking commands in Linux.	
6	Implement different topologies using Cisco packet tracer.	
7	Implement Unicast Routing Algorithm. (Distance Vector Routing Algorithm)	
8	Implement Unicast Routing Algorithm. (Link State Routing Algorithm)	
9	Implementation of Congestion Control Algorithm (Leaky Bucket Algorithm.)	
10	Implementation of Multicast Routing Algorithm.	
11	Implementation of Dijkstra's Shortest Path Algorithm.	
12	Simulation or Implementation of DHCP.	
13	Simulation or Implementation of FTP.	
14	Implementation of Client- server program using iterative TCP server.	

## **LABORATORY OUTCOMES**

The practical/exercises in this section are psychomotor domain Learning Outcomes (i.e., subcomponents of the COs), to be developed and assessed to lead to the attainment of the competency.

CO-1: Identify and use various networking components.

CO2 - Understand different transmission media and design cables for establishing a network.

CO-3: Use of basic networking commands in Windows/ Linux

CO-4: Able to design and implement various network applications such as FTP, DHCP.

CO-5: Understand the various routing protocols/ algorithms and internetworking.

**DOs and DON'Ts in Laboratory:**

1. Make an entry in the Log Book as soon as you enter the Laboratory.
2. All the students should sit according to their roll numbers starting from their left to right.
3. All the students are supposed to enter the terminal number in the log book.
4. Do not change the terminal on which you are working.
5. All the students are expected to get at least the algorithm of the program/concept to be implemented.
6. Strictly follow the instructions given by the teacher/Lab Instructor.

**Instruction for Laboratory Teachers**

1. Submissions related to whatever lab work has been completed should be done during the next lab session.
2. The immediate arrangements for printouts related to submission on the day of practical assignments.
3. Students should be taught to take the printouts under the observation of the lab teacher.
4. The promptness of submission should be encouraged by way of marking and evaluation patterns that will benefit the sincere students



## Lab Exercise :01

### Exercise No 1: ( 2 Hours) – 1 Practical

**AIM: Configure of network –Assigning IP Address, Subnetmask, gateway and Testing basic connectivity.**

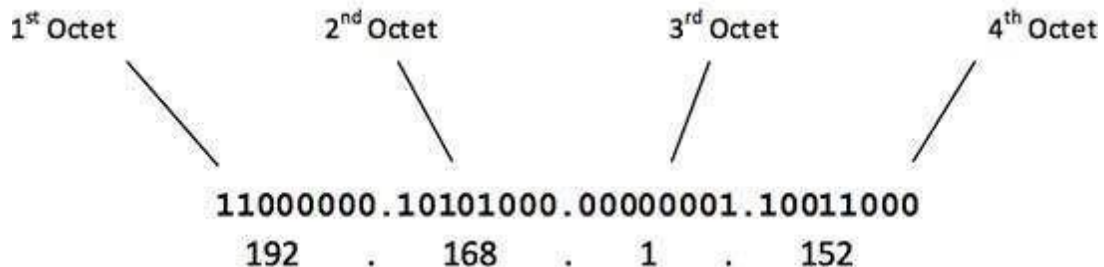
**Objective:** Students should be able to configure the network by assigning IP Address, Subnet mask, gateway and Testing Basic Connectivity.

**Theory:**

#### ✓ IP ADDRESS :

Internet Protocol hierarchy contains several classes of IP Addresses to be used efficiently in various situations as per the requirement of hosts per network. Broadly, the IPv4 Addressing system is divided into five classes of IP Addresses. All the five classes are identified by the first octet of IP Address. Internet Corporation for Assigned Names and Numbers is responsible for assigning IP addresses.

The first octet referred here is the left most of all. The octets numbered as follows depicting dotted decimal notation of IP Address:



The number of networks and the number of hosts per class can be derived by this formula:

Number of networks =  $2^{\text{network\_bits}}$

Number of Hosts/Network =  $2^{\text{host\_bits}} - 2$

When calculating hosts' IP addresses, 2 IP addresses are decreased because they cannot be assigned to hosts, i.e. the first IP of a network is network number and the last IP is reserved for Broadcast IP.

- **Class A Address:**

The first bit of the first octet is always set to 0 (zero). Thus the first octet ranges from 1 – 127, i.e.

**00000001 – 01111111**  
1 – 127

Class A addresses only include IP starting from 1.x.x.x to 126.x.x.x only. The IP range 127.x.x.x is reserved for loopback IP addresses.

The default subnet mask for Class A IP address is 255.0.0.0 which implies that Class A addressing can have 126 networks ( $2^7-2$ ) and 16777214 hosts ( $2^{24}-2$ ).

Class A IP address format is thus: 0NNNNNNN.HHHHHHHH.HHHHHHHH.HHHHHHHH

- **Class B Address:**

An IP address which belongs to class B has the first two bits in the first octet set to 10, i.e.

**10000000 – 10111111**  
128 – 191

Class B IP Addresses range from 128.0.x.x to 191.255.x.x. The default subnet mask for Class B is 255.255.x.x.

Class B has 16384 ( $2^{14}$ ) Network addresses and 65534 ( $2^{16}-2$ ) Host addresses. Class B IP address format is: 10NNNNNN.NNNNNNNN.HHHHHHHH.HHHHHHHH

- **Class C Address:**

The first octet of Class C IP address has its first 3 bits set to 110, that is:

**11000000 – 11011111**  
192 – 223

Class C IP addresses range from 192.0.0.x to 223.255.255.x. The default subnet mask for Class C is 255.255.255.x.

Class C gives 2097152 ( $2^{21}$ ) Network addresses and 254 ( $2^8-2$ ) Host addresses. Class C IP address format is: **110NNNNN.NNNNNNNN.NNNNNNNN.HHHHHHHH**

- **Class D Address:**

Very first four bits of the first octet in Class D IP addresses are set to 1110, giving a range of:

**11100000 – 11101111**  
**224 – 239**

Class D has IP address range from 224.0.0.0 to 239.255.255.255. Class D is reserved for Multicasting. In multicasting data is not destined for a particular host, that is why there is no need to extract host address from the IP address, and Class D does not have any subnet mask.

- **Class E Address:**

This IP Class is reserved for experimental purposes only for R&D or Study. IP addresses in this class ranges from 240.0.0.0 to 255.255.255.254. Like Class D, this class too is not equipped with any subnet mask.

Class	1st Octet Decimal Range	1st Octet High Order Bits	Network/Host ID (N=Network, H=Host)	Default Subnet Mask	Number of Networks	Hosts per Network (Usable Addresses)
A	1-126*	0	N.H.H.H	255.0.0.0	$126(2^7 - 2)$	$16,777,214 (2^{24} - 2)$
B	128-191	10	N.N.H.H	255.255.0.0	$16,382 (2^{14} - 2)$	$65,534 (2^{16} - 2)$
C	192-223	110	N.N.N.H	255.255.255.0	$2,097,150 (2^{21} - 2)$	$254(2^8 - 2)$
D	224-239	1110	Reserved for Multicasting			
E	240-254	11110	Experimental; used for research			

## SUBNET MASK:

The 32-bit IP address contains information about the host and its network. It is very necessary to distinguish both. For this, routers use Subnet Mask, which is as long as the size of the network address in the IP address. Subnet Mask is also 32 bits long. If the IP address in binary is ANDed with its Subnet Mask, the result yields the Network address. For example, say the IP Address is 192.168.1.152 and the Subnet Mask is 255.255.255.0 then:

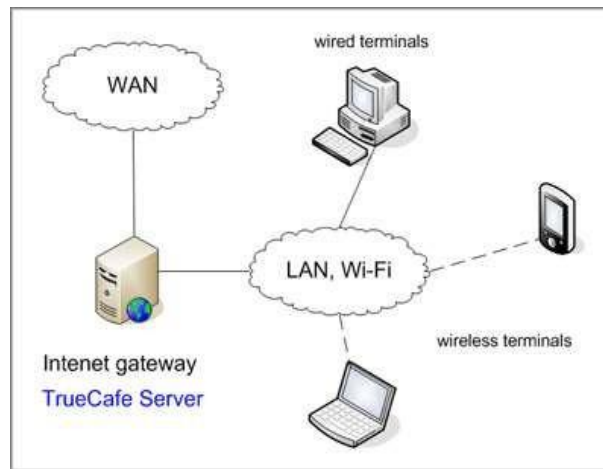
IP	192.168.1.152	11000000	10101000	00000001	10011000	} ANDed
Mask	255.255.255.0	11111111	11111111	11111111	00000000	
Network	192.168.1.0	11000000	10101000	00000001	00000000	Result

This way the Subnet Mask helps extract the Network ID and the Host from an IP Address. It can be identified now that 192.168.1.0 is the Network number and 192.168.1.152 is the host on that network.

## GATEWAY:

A **network gateway** is an internetworking system capable of joining together two networks that use different base protocols. A network gateway can be implemented completely in software, completely in hardware, or as a combination of both. Depending on the types of protocols they support, network gateways can operate at any level of the OSI model.

In enterprises, the gateway node often acts as a proxy server and firewall. The gateway is also associated with both a router, which use headers and forwarding tables to determine where packets are sent, and a switch, which provides the actual path for the packet in and out of the gateway. A computer system located on earth that switches data signals and voice signals between satellites and terrestrial networks.



## TESTING BASIC CONNECTIVITY:

### The ping Command

The basic format of the **ping** command on a Solaris system is: [2]

[2] Check your system's documentation. **ping** varies slightly from system to system. On Linux, the format shown above would be: **ping [-c count] [-s packetsize] host**

**ping** host [packetsize] [count]

host....

The hostname or IP address of the remote host being tested. Use the hostname or address provided by the user in the trouble report.

Packetsize.....

Defines the size in bytes of the test packets. This field is required only if the count field is going to be used. Use the default packetsize of 56 bytes.

Count.....

The number of packets to be sent in the test. Use the count field, and set the value low. Otherwise, the **ping** command may continue to send test packets until you interrupt it, usually by pressing CTRL-C (^C). Sending excessive numbers of test

packets is not a good use of network bandwidth and system resources. Usually five packets are sufficient for a test.

To check that ns.uu.net can be reached from almond, we send five 56-byte packets with the following command:

```
% ping -s ns.uu.net 56 5
```

```
PING ns.uu.net: 56 data bytes
```

```
64 bytes from ns.uu.net (137.39.1.3): icmp_seq=0. time=32.8 ms 64
bytes from ns.uu.net (137.39.1.3): icmp_seq=1. time=15.3 ms 64 bytes
from ns.uu.net (137.39.1.3): icmp_seq=2. time=13.1 ms 64 bytes from
ns.uu.net (137.39.1.3): icmp_seq=3. time=32.4 ms 64 bytes from
ns.uu.net (137.39.1.3): icmp_seq=4. time=28.1 ms
```

```
---ns.uu.net PING Statistics---
```

```
5 packets transmitted, 5 packets received, 0% packet loss round-
trip (ms) min/avg/max = 13.1/24.3/32.8
```

The **-s** option is included because almond is a Solaris workstation, and we want packet-by- packet statistics. Without the **-s** option, Sun's **ping** command only prints a summary line saying "ns.uu.net is alive." Other **ping** implementations do not require the **-s** option; they display the statistics by default.

The statistics displayed by the **ping** command can indicate low-level network problems. The key statistics are:

- The sequence in which the packets are arriving, as shown by the ICMP sequence number (icmp\_seq) displayed for each packet.
- How long it takes a packet to make the round trip, displayed in milliseconds after the string time=.
- The percentage of packets lost, displayed in a summary line at the end of the **ping** output.

If the packet loss is high, the response time is very slow, or packets are arriving out of order, there could be a network hardware problem. If you see these conditions when communicating over great distances on a wide area network, there is nothing to worry about. TCP/IP was designed to deal with unreliable networks, and some wide area networks suffer a lot of packet loss. But if these problems are seen on a local area network, they indicate trouble.

The results of a simple **ping** test, even if the **ping** is successful, can help you direct further testing toward the most likely causes of the problem. But other diagnostic tools are needed to examine the problem more closely and find the underlying cause.

### Conclusion:

IP Addresses to be used as per the requirement of hosts per network . Subnet Mask helps extract the Network ID and the Host from an IP Address.

A **network gateway** is an *internetworking* system capable of joining together two networks.

## Lab Exercise:02

Exercise No 2: ( 2 Hours) – 1 Practical

**AIM:** Study of Networking Device

**Objective:** To provide a basic understanding of how our computers are connected together via a network.

**Theory:-**Hardware devices that are used to connect computers, printers, fax machines and other electronic devices to a network are called network devices. These devices transfer data in a fast, secure and correct way over same or different networks. Network devices may be inter-network or intra -network.

### Types of Network Devices

There are different types of network devices used in a computer network which include the following.

- Network Hub
- Network Switch
- Modem
- Network Router
- Bridge
- Repeater

### Network Hub

The network hub is one kind of networking device in a computer network, used to communicate with various network hosts and also for data transferring. The transferring of data in a computer network can be done in the form of packets. Whenever the data processing can be done from a host to a network hub, then the data can transmit to all the connected ports. Similarly, all the ports identify the data path which leads to inefficiencies & wastage. Because of this working, a network hub cannot be so safe and secure. In addition, copying the data packets on all the ports will make the hub slower which leads to the utilize of the network switch.

Network hubs are classified into two types like active hub & passive hub.

### Active Hub

These hubs have their own power supply and these hubs are used to clean, increase & transmit the signal using the network. It works as a wiring center & repeater. Active hubs play a key role in extending the distance between nodes.

### Passive Hub

These hubs collect wiring from the power supply and different nodes of an active hub. These hubs transmit the signals over the network without improving & cleaning them. These hubs are not suitable for extending the distance between nodes like an active hub.

### Network Switch

Similar to a hub, this is also working at the layer in the LAN and a switch is more clever compare with a hub. As the hub is used for data transferring, whereas a switch is used for filtering & forwarding the data. So this is the more clever technique to deal with the data packets.

Whenever a data packet is obtained from the interfaces in the switch, then the data packet can be filtered & transmits to the interface of the proposed receiver. Due to this reason, a switch maintains a content addressable memory table to maintain system configuration as well as memory. This table is also named as FIB (forwarding information base) otherwise forwarding table.

### **Modem**

A modem is the most important network device and it is used daily in our life. If we notice the internet connection to homes was given with the help of a wire. then wire carries internet data from one place to another. But, every computer gives digital or binary data in the form of zeros & ones

The full form of the modem is a modulator and a demodulator. So it modulates as well as demodulates the signal among the computer and a telephone line because the computer generates digital data whereas the telephone line generates an analog signal.

### **Network Router**

A network router is one kind of network device in a computer network and it is used for routing traffic from one network to another. These two networks could be private to a public company network. For example, here a router is considered as traffic police at the junction, he directs dissimilar traffic networks to dissimilar directions.

### **Bridge**

A Bridge in the computer network is used to unite two or more network segments. The main function of a bridge in network architecture is to store as well as transmit frames among the various segments. Bridges use MAC (Media Access Control) hardware for transferring frames. These are also used for connecting two physical local area networks to a larger logical local area network. In the OSI model, bridges work at the data link & physical layers to divide the networks from larger to smaller by controlling the data flow between the two. In recent years, bridges are replaced by switches to provide more functionality.

### **Repeater**

The operating of a repeater can be done at the physical layer. The main function of this device is to reproduce the signal on a similar network before the signal gets weak otherwise damaged. The significant point to be noted regarding these devices is that they do not strengthen the signal. Whenever the signal gets weak, then they reproduce it at the actual strength. A repeater is a two-port device.

### **Gateway**

Generally, a gateway performs at the session & transport layers in the OSI model. Gateways offer conversion between networking technologies like OSI (Open System Interconnection) & TCP/IP. Because of this, these are connected to two or many autonomous networks, where each network has its own domain name service, routing algorithm, topology, protocols, and procedures of network administration & policies Gateways execute all the functions of routers. Actually, a router with additional conversion functionality is a gateway, so the conversion between various network technologies is known as a protocol converter.

### **Router**

The router is also called a bridging router and the main function of this is to combine the features of both router & bridge and router. It performs either at the network layer or the data link layer. When it works as a



router, it is used for routing packets across networks whereas it works as a bridge; it is used for filtering LAN's traffic.

**Conclusion:** -A network is two or more computers connected together using a telecommunication system for the purpose of communicating and sharing resources.

## Lab Exercise: 03

Exercise No 2: ( 2 Hours) – 1 Practical

**Title:** To study Transmission media

**Objective:**

To study the various transmission media used for electrical communication.

To study the characteristics and advantages of each for practical communication systems.

**Theory:**

A **transmission medium** is something that can mediate the propagation of signals for the purposes of telecommunication. Signals are typically imposed on a wave of some kind suitable for the chosen medium. For example, data can modulate sound, and a transmission medium for sounds may be air, but solids and liquids may also act as the transmission medium. Vacuum or air constitutes a good transmission medium for electromagnetic waves such as light and radio waves. While material substance is not required for electromagnetic waves to propagate, such waves are usually affected by the transmission media they pass through, for instance, by absorption or reflection or refraction at the interfaces between media. Technical devices can therefore be employed to transmit or guide waves. Thus, an optical fiber or a copper cable is used as transmission media.

### **Types of Transmission Media**

In data communication terminology, a transmission medium is a physical path between the transmitter and the receiver i.e. it is the channel through which data is sent from one place to another. Transmission Media is broadly classified into the following types:

#### **1. Guided Media:**

It is also referred to as Wired or Bounded transmission media. Signals being transmitted are directed and confined in a narrow pathway by using physical links.

Features:

- High Speed
- Secure
- Used for comparatively shorter distances

There are 3 major types of Guided Media:

#### **(i) Twisted Pair Cable –**

It consists of 2 separately insulated conductor wires wound about each other. Generally, several such pairs are bundled together in a protective sheath. They are the most widely used Transmission Media. Twisted Pair is of two types:

##### **1. Unshielded Twisted Pair (UTP):**

This type of cable has the ability to block interference and does not depend on a physical shield for this purpose. It is used for telephonic applications.

Advantages:

- Least expensive
- Easy to install
- High-speed capacity
- Susceptible to external interference
- Lower capacity and performance in comparison to STP
- Short distance transmission due to attenuation

### 3. **Shielded Twisted Pair (STP):**

This type of cable consists of a special jacket to block external interference. It is used in fast-data-rate Ethernet and in voice and data channels of telephone lines.

Advantages:

4.

- Better performance at a higher data rate in comparison to UTP
- Eliminates crosstalk
- Comparatively faster
- Comparatively difficult to install and manufacture
- More expensive
- Bulky

### **(ii) Coaxial Cable –**

It has an outer plastic covering containing 2 parallel conductors each having a separate insulated protection cover. The coaxial cable transmits information in two modes: Baseband mode(dedicated cable bandwidth) and Broadband mode(cable bandwidth is split into separate ranges). Cable TVs and analog television networks widely use Coaxial cables.

Advantages:

- High Bandwidth
- Better noise Immunity
- Easy to install and expand
- Inexpensive

Disadvantages:

- Single cable failure can disrupt the entire network

### **(iii) Optical Fibre Cable –**

It uses the concept of reflection of light through a core made up of glass or plastic. The core is surrounded by a less dense glass or plastic covering called the cladding. It is used for the transmission of large volumes of data.

The cable can be unidirectional or bidirectional. The WDM (Wavelength Division Multiplexer) supports two modes, namely unidirectional and bidirectional mode.

Advantages:

- Increased capacity and bandwidth
- Lightweight
- Less signal attenuation
- Immunity to electromagnetic interference
- Resistance to corrosive materials

Disadvantages:

- Difficult to install and maintain
- High cost
- Fragile
- 

#### **(iv) Stripline**

Stripline is a transverse electromagnetic (TEM) transmission line medium invented by Robert M. Barrett of the Air Force Cambridge Research Centre in the 1950s. Stripline is the earliest form of the planar transmission line. It uses a conducting material to transmit high-frequency waves it is also called a waveguide. This conducting material is sandwiched between two layers of the ground plane which are usually shorted to provide EMI immunity.

#### **(v) Microstripline**

In this, the conducting material is separated from the ground plane by a layer of dielectric.

### **2. Unguided Media:**

It is also referred to as Wireless or Unbounded transmission media. No physical medium is required for the transmission of electromagnetic signals.

Features:

- The signal is broadcasted through air
- Less Secure
- Used for larger distances

There are 3 types of Signals transmitted through unguided media:

#### **(i) Radiowaves –**

These are easy to generate and can penetrate through buildings. The sending and receiving antennas need not be aligned. Frequency Range: 3KHz – 1GHz. AM and FM radios and cordless phones use Radiowaves for transmission.

Further Categorized as (i) Terrestrial and (ii) Satellite

#### **(ii) Microwaves –**

It is a line of sight transmission i.e. the sending and receiving antennas need to be properly aligned with each other. The distance covered by the signal is directly proportional to the height of the antenna. Frequency Range: 1GHz – 300GHz. These are majorly used for mobile phone communication and television distribution.

#### **(iii) Infrared –**

Infrared waves are used for very short distance communication. They cannot penetrate through obstacles. This prevents interference between systems. Frequency Range: 300GHz – 400THz. It is used in TV remotes, wireless mouse, keyboard, printer, etc.

**Conclusion:** In summary, transmission media are essential for communication systems. Signals that carry information can be transmitted on a transmission medium for communication purposes. The transmission characteristics of the medium in use are important because they directly affect the communication quality.

## Lab Exercise: 04

Exercise No 2: ( 2 Hours) – 1 Practical

Title: Creating a networking cable using crimping tool & Study different connectors

Objective: Study different types of networking cables, connectors and other network components.

Theory:

There are three specific types network cables, and the connectors associated with each, that you must know for this exam: fiber, twisted pair, and coaxial. ... The two most regularly used connectors are F-connectors (television cabling) and BNC (10Base2, and so on).

Networking cables are networking hardware used to connect one network device to other network devices or to connect two or more computers to share printers, scanners etc. Different types of network cables, such as coaxial cable, optical fiber cable, and twisted pair cables, are used depending on the network's physical layer, topology, and size. The devices can be separated by a few meters (e.g. via Ethernet) or nearly unlimited distances (e.g. via the interconnections of the Internet).

There are several technologies used for network connections. Patch cables are used for short distances in offices and wiring closets. Electrical connections using twisted pair or coaxial cable are used within a building. Optical fiber cable is used for long distances or for applications requiring high bandwidth or electrical isolation. Many installations use structured cabling practices to improve reliability and maintainability. In some home and industrial applications power lines are used as network cabling.

### Twisted Pair

Twisted pair cabling is a form of wiring in which pairs of wires (the forward and return conductors of a single circuit) are twisted together for the purposes of canceling out electromagnetic interference (EMI) from other wire pairs and from external sources. This type of cable is used for home and corporate Ethernet networks. Twisted pair cabling is used in short patch cables and in the longer runs in structured cabling.

An Ethernet crossover cable is a type of twisted pair Ethernet cable used to connect computing devices together directly that would normally be connected via a network switch, Ethernet hub or router, such as directly connecting two personal computers via their network adapters. Most current Ethernet devices support Auto MDI-X, so it doesn't matter whether you use crossover or straight cables.



## **Fiber Optics**

An optical fiber cable consists of a center glass core surrounded by several layers of protective material. Optical fiber deployment is more expensive than copper but offers higher bandwidth and can cover longer distances.[2] There are two major types of optical fiber cables: shorter-range multi-mode fiber and long-range single-mode fiber.

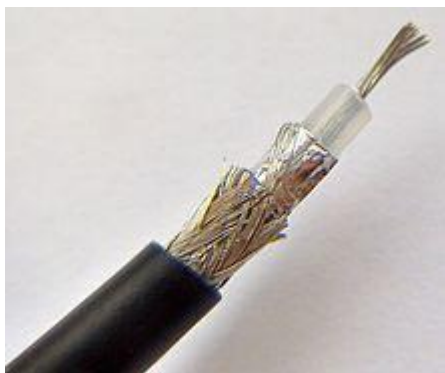


## **Coaxial**

Coaxial cables form a transmission line and confine the electromagnetic wave inside the cable between the center conductor and the shield. The transmission of energy in the line occurs totally through the dielectric inside the cable between the conductors. Coaxial lines can therefore be bent and twisted (subject to limits) without negative effects, and they can be strapped to conductive supports without inducing unwanted currents in them.

Early Ethernet, 10BASE5 and 10BASE2, used baseband signaling over coaxial cables. In the 20th century the L-carrier system used coaxial cable for long-distance calling.

Coaxial cables are commonly used for television and other broadband signals. Although in most homes coaxial cables have been installed for transmission of TV signals, new technologies (such as the ITU-T G.hn standard) open the possibility of using home coaxial cable for high-speed home networking applications (Ethernet over coax).



Conclusion: In this experiment we have studied that various networking cable using crimping tool.

## Lab Exercise: 05

Exercise No 2: ( 2 Hours) – 1 Practical

**Title:** Use basic networking commands.

**Objective:**

- To get familiar with different networking commands.

**Theory:**

1 - Tracert / traceroute :

Determines the path taken to a destination by sending Internet Control Message Protocol (ICMP) Echo Request messages to the destination with incrementally increasing Time to Live (TTL) field values. The path displayed is the list of near-side router interfaces of the routers in the path between a source host and a destination. The near-side interface is the interface of the router that is closest to the sending host in the path. Used without parameters, tracert displays help.

This diagnostic tool determines the path taken to a destination by sending ICMP Echo Request messages with varying Time to Live (TTL) values to the destination. Each router along the path is required to decrement the TTL in an IP packet by at least 1 before forwarding it.

Effectively, the TTL is a maximum link counter. When the TTL on a packet reaches 0, the router is expected to return an ICMP Time Exceeded message to the source computer. Tracert determines the path by sending the first Echo Request message with a TTL of 1 and incrementing the TTL by 1 on each subsequent transmission until the target responds or the maximum number of hops is reached. The maximum number of hops is 30 by default and can be specified using the -h parameter.

The path is determined by examining the ICMP Time Exceeded messages returned by intermediate routers and the Echo Reply message returned by the destination. However, some routers do not return Time Exceeded messages for packets with expired TTL values and are invisible to the tracert command. In this case, a row of asterisks (\*) is displayed for that hop.

Examples:

To trace the path to the host named www.google.co.in use following command

```
tracert www.google.co.in
```

To trace the path to the host named www.google.com and prevent the resolution of each IP address to its name, type:

```
tracert -d www.google.com
```

To trace the path to the host named www.google.com and use the loose source route 10.12.0.1-10.29.3.1-10.1.44.1, type:



```
tracert -j 10.12.0.1 10.29.3.1 10.1.44.1 www.google.com
```

## 2 - Ping :

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.

You can use ping to test both the computer name and the IP address of the computer. If pinging the IP address is successful, but pinging the computer name is not, you might have a name resolution problem. In this case, ensure that the computer name you are specifying can be resolved through the local Hosts file, by using Domain Name System (DNS) queries, or through NetBIOS name resolution techniques.

To test a TCP/IP configuration by using the ping command:

- To quickly obtain the TCP/IP configuration of a computer, open Command Prompt, and then type ipconfig . From the display of the ipconfig command, ensure that the network adapter for the TCP/IP configuration you are testing is not in a Media disconnected state.
- At the command prompt, ping the loopback address by typing ping 127.0.0.1
- Ping the IP address of the computer.
- Ping the IP address of the default gateway. If the ping command fails, verify that the default gateway IP address is correct and that the gateway (router) is operational.
- Ping the IP address of a remote host (a host that is on a different subnet). If the ping command fails, verify that the remote host IP address is correct, that the remote host is operational, and that all of the gateways (routers) between this computer and the remote host are operational.
- Ping the IP address of the DNS server. If the ping command fails, verify that the DNS server IP address is correct, that the DNS server is operational, and that all of the gateways (routers) between this computer and the DNS server are operational.

## 3 - Arp :

Displays and modifies entries in the Address Resolution Protocol (ARP) cache, which contains one or more tables that are used to store IP addresses and their resolved Ethernet or Token Ring physical addresses. There is a separate table for each Ethernet or Token Ring network adapter installed on your computer.

Syntax

```
arp [-a [InetAddr] [-N IfaceAddr]] [-g [InetAddr] [-N IfaceAddr]] [-d InetAddr [IfaceAddr]] [-s InetAddrEtherAddr [IfaceAddr]]
```

## 4 - Netstat :

Displays active TCP connections, ports on which the computer is listening, Ethernet statistics, the IP routing table, IPv4 statistics (for the IP, ICMP, TCP, and UDP protocols), and IPv6 statistics (for the IPv6, ICMPv6, TCP over IPv6, and UDP over IPv6 protocols).

Netstat provides statistics for the following:

- Proto - The name of the protocol (TCP or UDP).
- Local Address - The IP address of the local computer and the port number being used. The name of the local computer that corresponds to the IP address and the name of the port is shown unless the -n parameter is specified. If the port is not yet established, the port number is shown as an asterisk (\*).
- Foreign Address - The IP address and port number of the remote computer to which the socket is connected. The names that corresponds to the IP address and the port are shown unless the -n parameter is specified. If the port is not yet established, the port number is shown as an asterisk (\*).

(state) Indicates the state of a TCP connection. The possible states are as follows:

- CLOSE\_WAIT
- CLOSED
- ESTABLISHED
- FIN\_WAIT\_1
- FIN\_WAIT\_2
- LAST\_ACK
- LISTEN
- SYN\_RECEIVED
- SYN\_SEND
- TIMED\_WAIT

## Syntax

Netstat [-a] [-e] [-n] [-o] [-p Protocol] [r] [-s] [Interval]

## 5 - Ipconfig :

Displays all current TCP/IP network configuration values and refreshes Dynamic Host Configuration Protocol (DHCP) and Domain Name System (DNS) settings. This command is most useful on computers that are configured to obtain an IP address automatically. This enables users to determine which TCP/IP configuration values have been configured by DHCP, Automatic Private IP Addressing (APIPA), or an alternate configuration.

- If the Adapter name contains any spaces, use quotation marks around the adapter name (that is, "Adapter Name").
- For adapter names, ipconfig supports the use of the asterisk (\*) wildcard character to specify either adapters with names that begin with a specified string or adapters with names that contain a specified string.
- For example, **Local\*** matches all adapters that start with the string Local and **\*Con\*** matches all adapters that contain the string Con.

## Syntax

**Ipconfig** [/all] [/renew [Adapter]] [/release [Adapter]] [/flushdns] [/displaydns] [/registerdns] [/showclassid Adapter] [/setclassid Adapter [ClassID]]

## 6 - Winipcfg :

This utility allows users or administrators to see the current IP address and other useful information about your network configuration. You can reset one or more IP addresses. The Release or Renew buttons allow you to release or renew one IP address. If you want to release or renew all IP addresses click Release All or Renew All. When one of these buttons is clicked, a new IP address is obtained from either the DHCP service or from the computer assigning itself an automatic private IP address.

### To use the winipcfg utility:

- Click Start, and then click Run and type **winipcfg**
- Click More Info.
- To see the addresses of the DNS servers the computer is configured to use, click the ellipsis (...) button to the right of DNS Servers.
- To see address information for your network adapter(s), select an adapter from the list in Ethernet Adapter Information.

## 7 - Nslookup :

Nslookup (Name Server lookup) is a UNIX shell command to query Internet domain name servers.

### Definitions

- Nameserver: These are the servers that the internet uses to find out more about the domain. Usually they are an ISP's computer.
- Mailserver: Where email is sent to.
- Webserver: The domains website.
- FTPserver: FTP is file transfer protocol, this server is where files may be stored.
- Hostname: The name of the host as given by the domain.
- Real Hostname: This is hostname that you get by reverse resolving the IP address, may be different to the given hostname.
- IP Address: Unique four numbered identifier that is obtained by resolving the hostname.

### **Conclusion :**

Hence, we studied different networking commands.

## Lab Exercise: 06

Exercise No 2: ( 2 Hours) – 1 Practical

**Title:** To study different topologies in Cisco packet tracer

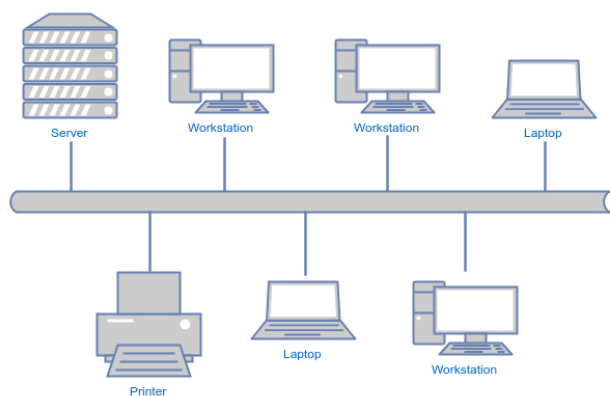
**Objective:** Study various topologies.

**Theory:-**

Network topology is the geometric representation of relationship of all the links connecting the devices or nodes. Network topology represent in two ways one is physical topology that define the way in which a network is physically laid out and other one is logical topology that defines how data actually flow through the network. In this paper we have discuss how to design bus, star and mesh topology network and provide interfacing and simulation between end points using packet tracer software.

**Bus Topology:-**

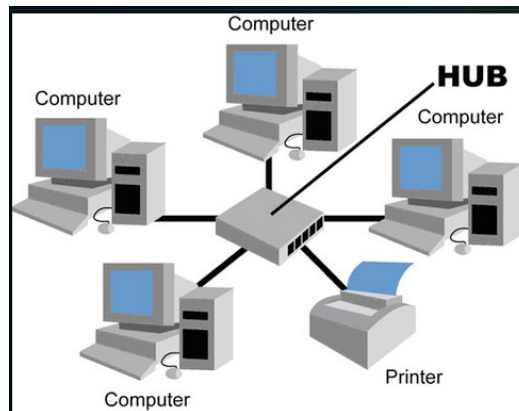
Bus Topology In local area network, it is a single network cable runs in the building or campus and all nodes are connected along with this communication line with two endpoints called the bus or backbone. In other words, it is a multipoint data communication circuit that is easily control data flow between the computers because this configuration allows all stations to receive every transmission over the network. For bus topology we build network using three generic pc which are serially connected with three switches using copper straight through cable and switches are interconnected using copper cross over cable.



***Bus Topology Network***

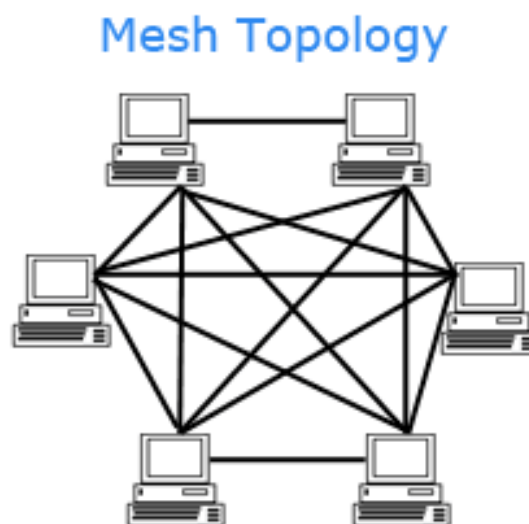
**Star Topology:-**

In star topology, all the cables run from the computers to a central location where they are all connected by a device called a hub. It is a concentrated network, where the end points are directly reachable from a central location when network is expanded. Ethernet 10 base T is a popular network based on the star topology. For star topology we build network using five generic pc which are centrally connected to single switch 2950-24 using copper straight through cable.



### Mesh Topology:-

In mesh topology every device has a dedicated point to point link to every other device. The term dedicated stand for link carries traffic only between two devices it connects. It is a well-connected topology; in this every node has a connection to every other node in the network. The cable requirements are high and it can include multiple topologies. Failure in one of the computers star topology, all the cables run from the computers to a central location.



### Ring Topology:-

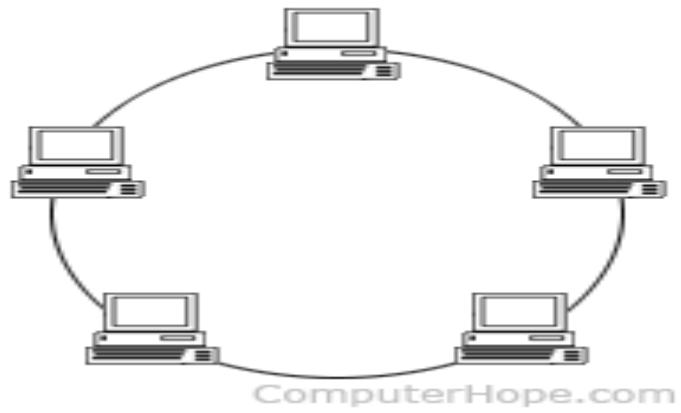
A **ring topology** is a network configuration where device connections create a circular data path. Each networked device is connected to two others, like points on a circle. Together, devices in a ring topology are referred to as a **ring network**.

In a ring network, packets of data travel from one device to the next until they reach their destination. Most ring topologies allow packets to travel only in one direction, called a **unidirectional** ring network. Others permit data to move in either direction, called **bidirectional**.

The major disadvantage of a ring topology is that if any individual connection in the ring is broken, the entire network is affected.

Ring topologies may be used in either LANs (local area networks) or WANs (wide area networks). Depending on the network card used in each computer of the ring topology, a coaxial cable or an RJ-45 network cable is used to connect computers together.

## Ring Topology



### **Conclusion:**

In this experiment we have to studied that various topologies and implementation.

## Lab Exercise: 07

Exercise No 2: ( 2 Hours) – 1 Practical

**AIM: Implement Unicast Routing Algorithm. (Distance Vector Routing Algorithm).**

**Objective:** Students should be able to develop a program on Distance Vector Routing Algorithm.

**Theory:**

✓ **Distance vector :**

Routing algorithm is a part of network layer software which is responsible for deciding which output line an incoming packet should be transmitted on. If the subnet uses datagram internally, this decision must be made anew for every arriving data packet since the best route may have changed since last time. If the subnet uses virtual circuits internally, routing decisions are made only when a new established route is being set up. The latter case is sometimes called session routing, because a route remains in force for an entire user session (e.g., login session at a terminal or a file).

Routing algorithms can be grouped into two major classes: adaptive and nonadaptive. Non adaptive algorithms do not base their routing decisions on measurement or estimates of current traffic and topology. Instead, the choice of route to use to get from  $I$  to  $J$  (for all  $I$  and  $J$ ) is computed in advance, offline, and downloaded to the routers when the network is booted. This procedure is sometimes called static routing.

Adaptive algorithms, in contrast, change their routing decisions to reflect changes in the topology, and usually the traffic as well. Adaptive algorithms differ in where they get information (e.g., locally, from adjacent routers, or from all routers), when they change the routes (e.g., every  $\Delta T$  sec, when the load changes, or when the topology changes), and what metric is used for optimization (e.g., distance, number of hops, or estimated transit time).

Two algorithms in particular, distance vector routing and link state routing are the most popular. Distance vector routing algorithms operate by having each router maintain a table (i.e., vector) giving the best known distance to each destination and which line to get there. These tables are updated by exchanging information with the neighbors.

The distance vector routing algorithm is sometimes called by other names, including the distributed Bellman-Ford routing algorithm and the Ford-Fulkerson algorithm, after the researchers who developed it (Bellman, 1957; and Ford and Fulkerson, 1962). It was the original ARPANET routing algorithm and was also used in the Internet under the RIP and in early versions of DECnet and Novell's IPX. AppleTalk and Cisco routers use improved distance vector protocols.

In distance vector routing, each router maintains a routing table indexed by, and containing one entry for, each router in the subnet. This entry contains two parts: the preferred outgoing line to use for that destination, and an estimate of the time or distance to that destination. The metric used might be number of hops, time delay in milliseconds, total number of packets queued along the path, or something similar.

The router is assumed to know the “distance” to each of its neighbor. If the metric is hops, the distance is just one hop. If the metric is queue length, the router simply examines each queue. If the metric is delay, the router can measure it directly with special ECHO packets that the receiver just time stamps and sends back as fast as possible.

### **The Count to Infinity Problem.**

Distance vector routing algorithm reacts rapidly to good news, but leisurely to bad news. Consider a router whose best route to destination  $X$  is large. If on the next exchange neighbor  $A$  suddenly reports a short delay to  $X$ , the router just switches over to using the line to  $A$  to send traffic to  $X$ . In one vector exchange, the good news is processed.

To see how fast good news propagates, consider the five node (linear) subnet of the following figure, where the delay metric is the number of hops. Suppose  $A$  is down initially and all the other routers know this. In other words, they have all recorded the delay to  $A$  as infinity.

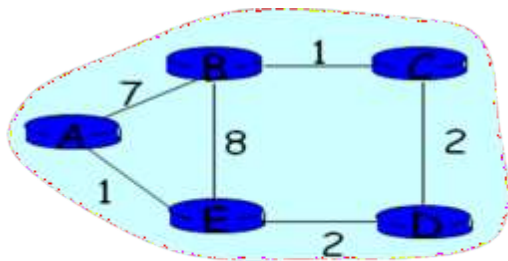
A      B      C      D      E                      A      B      C      D      E

$\infty$	$\infty$	$\infty$	$\infty$	Initially	1	2	3	4	Initially
1	$\infty$	$\infty$	$\infty$	After 1 exchange	3	2	3	4	After 1 exchange
1	2	$\infty$	$\infty$	After 2 exchange	3	3	3	4	After 2 exchange
1	2	3	$\infty$	After 3 exchange	5	3	5	4	After 3 exchange
1	2	3	4	After 4 exchange	5	6	5	6	After 4 exchange
					7	6	7	6	After 5 exchange
					7	8	7	8	After 6 exchange
						:			
					$\infty$	$\infty$	$\infty$	$\infty$	

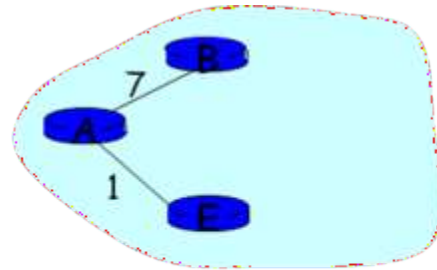


Many ad hoc solutions to the count to infinity problem have been proposed in the literature, each one more complicated and less useful than the one before it. The **split horizon** algorithm works the same way as distance vector routing, except that the distance to  $X$  is not reported on line that packets for  $X$  are sent on (actually, it is reported as infinity). In the initial state of the right figure, for example,  $C$  tells  $D$  the truth about distance to  $A$  but  $C$  tells  $B$  that its distance to  $A$  is infinite. Similarly,  $D$  tells the truth to  $E$  but lies to  $C$ .

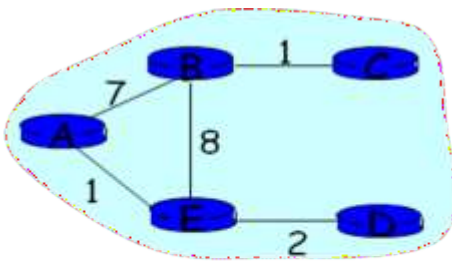
- A's distance vector  $D(A,*)$ :
  - After Iteration 1 is: [0, 7, INFINITY, INFINITY, 1]
  - After Iteration 2 is: [0, 7, 8, 3, 1]
  - After Iteration 3 is: [0, 7, 5, 3, 1]
  - After Iteration 4 is: [0, 6, 5, 3, 1]



**Example network**



**A's 1-hop view  
(After 1<sup>st</sup> iteration)**



**A's 2-hop view  
(After 2<sup>nd</sup> Iteration)**

### Algorithm:

- $c(x,v)$  = cost for direct link from x to v
- Node x maintains costs of direct links  $c(x,v)$
- $D_x(y)$  = estimate of least cost from x to y
- Node x maintains distance vector  $\mathbf{D}_x = [D_x(y): y \in N]$
- Node x maintains its neighbors' distance vectors
- For each neighbor v, x maintains  $\mathbf{D}_v = [D_v(y): y \in N]$
- Each node v periodically sends  $D_v$  to its neighbors
- And neighbors update their own distance vectors
- $D_x(y) \leftarrow \min_v \{c(x,v) + D_v(y)\}$  for each node  $y \in N$
- Over time, the distance vector  $D_x$  converges

### Output:

A sample run of the program works as:-

Enter the number of nodes :

3

Enter the cost matrix :

0 2 7

2 0 1

7 1 0

For router 1

node 1 via 1 Distance 0

node 2 via 2 Distance 2

node 3 via 3 Distance 3

For router 2

node 1 via 1 Distance 2

node 2 via 2 Distance 0

node 3 via 3 Distance 1

For router 3

node 1 via 1 Distance 3

node 2 via 2 Distance 1

node 3 via 3 Distance 0

**Conclusion:**

The least-cost route between any two nodes is the route with minimum distance.

Each node maintains a vector of minimum distances to every node

## Lab Exercise: 08

Exercise No 3: ( 2 Hours) – 1 Practical

**AIM: Implement Unicast Routing Algorithm. (Link State Routing Algorithm)**

**Objective:** Students should be able to develop a program on Link State Routing Algorithm.

**Theory:**

**Link State Routing:**

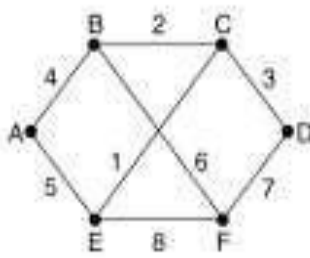
A link-state routing protocol is one of the two main classes of routing protocols used in packet switching networks for computer communications (the other is the distance-vector routing protocol). Examples of link-state routing protocols include open shortest pathfirst (OSPF) and intermediate system to intermediate system.

The link-state protocol is performed by every switching node in the network (i.e., nodes that are prepared to forward packets; in the Internet, these are called routers). The basic concept of link-state routing is that every node constructs a map of the connectivity to the network, in the form of a graph, showing which nodes are connected to which other nodes. Each node then independently calculates the next best logical path from it to every possible destination in the network. The collection of best paths will then form the node's routing table.

This contrasts with distance-vector routing protocols, which work by having each node share its routing table with its neighbors. In a link-state protocol the only information passed between nodes is connectivity related. Distance vector routing was used in the ARPANET until 1979, when it was replaced by link state routing. Two primary problems caused its demise. First, since the delay metric was queue length, it did not take line bandwidth into account when choosing routes. Initially, all the lines were 56 kbps, so line bandwidth was not an issue, but after some lines had been upgraded to 230 kbps and others to 1.544 Mbps, not taking bandwidth into account was a major problem. Of course, it would have been possible to change the delay metric to factor in line bandwidth, but a second problem also existed, namely, the algorithm often took too long to converge (the count-to-infinity problem).

The idea behind link state routing is simple and can be stated as five parts. Each router must do the following:

1. Discover its neighbors and learn their network addresses.
2. Measure the delay or cost to each of its neighbors.
3. Construct a packet telling all it has just learned.
4. Send this packet to all other routers.
5. Compute the shortest path to every other router.



(a)

Link		State		Packets	
A	B	C	D	E	F
Seq.	Seq.	Seq.	Seq.	Seq.	Seq.
Age	Age	Age	Age	Age	Age
B 4	A 4	B 2	C 3	A 5	B 6
E 5	C 2	D 3	F 7	C 1	D 7
	F 6	E 1		F 8	E 8

(b)

### 1. Learning about the Neighbors:

When a router is booted, its first task is to learn who its neighbors are. It accomplishes this goal by sending a special HELLO packet on each point-to-point line. The router on the other end is expected to send back a reply telling who it is. These names must be globally unique because when a distant router later hears that three routers are all connected to F, it is essential that it can determine whether all three mean the same F.

### 2. Measuring Line Cost:

The link state routing algorithm requires each router to know, or at least have a reasonable estimate of, the delay to each of its neighbors. The most direct way to determine this delay is to send over the line a special ECHO packet that the other side is required to send back immediately. By measuring the round-trip time and dividing it by two, the sending router can get a reasonable estimate of the delay. For even better results, the test can be conducted several times, and the average used. Of course, this method implicitly assumes the delays are

symmetric, which may not always be the case. An interesting issue is whether to take the load into account when measuring the delay. To factor the load in, the round-trip timer must be started when the ECHO packet is queued. To ignore the load, the timer should be started when the ECHO packet reaches the front of the queue.

### **3. Building Link State Packets:**

Once the information needed for the exchange has been collected, the next step is for each router to build a packet containing all the data. The packet starts with the identity of the sender, followed by a sequence number and age (to be described later), and a list of neighbors. For each neighbor, the delay to that neighbor is given.

### **4. Distributing the Link State Packets:**

The trickiest part of the algorithm is distributing the link state packets reliably. As the packets are distributed and installed, the routers getting the first ones will change their routes. Consequently, the different routers may be using different versions of the topology, which can lead to inconsistencies, loops, unreachable machines, and other problems.

First we will describe the basic distribution algorithm. Later we will give some refinements. The fundamental idea is to use flooding to distribute the link state packets. To keep the flood in check, each packet contains a sequence number that is incremented for each new packet sent. Routers keep track of all the (source router, sequence) pairs they see. When a new link state packet comes in, it is checked against the list of packets already seen. If it is new, it is forwarded on all lines except the one it arrived on. If it is a duplicate, it is discarded. If a packet with a sequence number lower than the highest one seen so far ever arrives, it is rejected as being obsolete since the router has more recent data.

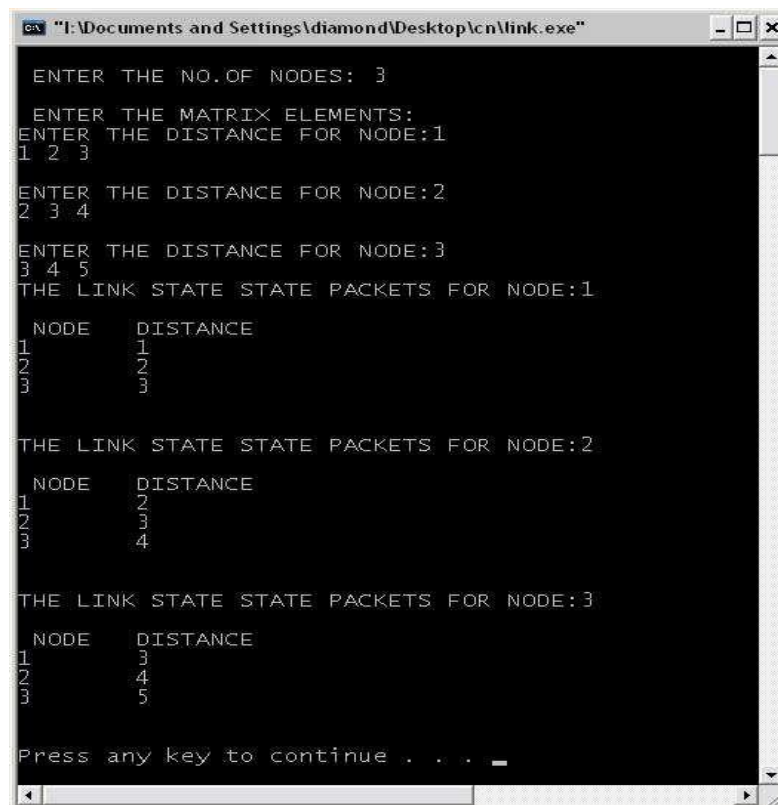
### **5. Computing the New Routes:**

Once a router has accumulated a full set of link state packets, it can construct the entire subnet graph because every link is represented. Every link is, in fact, represented twice, once for each direction. The two values can be averaged or used separately. Now Dijkstra's algorithm can be run locally to construct the shortest path to all possible destinations. The results of this algorithm can be installed in the routing tables, and normal operation resumed. For a subnet with  $n$  routers, each of which has  $k$  neighbors, the memory required to store the input data is proportional to  $kn$ . For large subnets, this can be a problem. Also, the computation time can be an issue. Nevertheless, in many practical situations, link state routing works well.

### Algorithm:

```
1   Initialization:
2   N' = {u}
3   for all nodes v
4   if v adjacent to u
5   then D(v) = c(u,v)
6   else D(v) = ∞
7   Loop
8   find w not in N' such that D(w) is a minimum
9   add w to N'
10  update D(v) for all v adjacent to w and not in N' : 12
    D(v) = min( D(v), D(w) + c(w,v) )
13  /* new cost to v is either old cost to v or known
14  shortest path cost to w plus cost from w to v */
15  until all nodes in N'
```

### Output:



```
"I:\Documents and Settings\diamond\Desktop\cn\link.exe"

ENTER THE NO.OF NODES: 3

ENTER THE MATRIX ELEMENTS:
ENTER THE DISTANCE FOR NODE:1
1 2 3

ENTER THE DISTANCE FOR NODE:2
2 3 4

ENTER THE DISTANCE FOR NODE:3
3 4 5

THE LINK STATE STATE PACKETS FOR NODE:1

NODE    DISTANCE
1        1
2        2
3        3

THE LINK STATE STATE PACKETS FOR NODE:2

NODE    DISTANCE
1        2
2        3
3        4

THE LINK STATE STATE PACKETS FOR NODE:3

NODE    DISTANCE
1        3
2        4
3        5

Press any key to continue . . .
```

### Conclusion:

Each router keeps track of its incident links .Each router runs Dijkstra's algorithm.

## Lab Exercise: 09

Exercise No 4: ( 2 Hours) – 1 Practical

**AIM:- Implementation of Multicast Routing Algorithm.**

**Objective:** Student should be able to develop a program on Multicast or Broadcast routing. It is defined in the form of minimizing the cost of a multicast tree

### **Theory:**

#### **Multicast:**

Multicast is communication between a single sender and multiple receivers on a network. Typical uses include the updating of mobile personnel from a home office and the periodic issuance of online newsletters. Together with anycast and unicast, multicast is one of the packet types in the Internet Protocol Version 6 (IPv6).

#### **MulticastSocket**

```
public MulticastSocket()  
    throws IOException  
Create a multicast socket.
```

If there is a security manager, its `checkListen` method is first called with 0 as its argument to ensure the operation is allowed. This could result in a `SecurityException`.

Throws: `IOException` - if an I/O exception occurs while creating the  
`MulticastSocketSecurityException` - if a security manager exists and its `checkListen` method doesn't allow the operation.

```
public MulticastSocket(int port)  
    throws IOException
```

Create a multicast socket and bind it to a specific port.

If there is a security manager, its `checkListen` method is first called with the port argument as its argument to ensure the operation is allowed. This could result in a `SecurityException`.

When the socket is created the `DatagramSocket.setReuseAddress(boolean)` method is called to enable the `SO_REUSEADDR` socket option.



Parameters:

port - port to use

**Algorithm:**

**Pseudo code for multicast source:**

```
import java.io.*;
import java.net.*;
public class msource
{
    public static void main(String args[])
    {
        try
        {
            DatagramSocket s=new DatagramSocket(); byte[]
            smsg=new byte[100]; System.out.println("Enter
            the text to send : "); int len=System.in.read(smsg);
            InetAddress test=InetAddress.getLocalHost();

            DatagramPacket pack=new DatagramPacket(smsg,len,test,16900); s.send(pack);
lose();
        }
        catch(Exception err)
        System.out.println(err);
    }
}
}
```

**Pseudo code for multicast source:**

```
import java.io.*;
import java.net.*;
public class mulcli
```

```

{
public static void main(String args[])
{
try
{
MulticastSocket mul=new MulticastSocket(16900);
mul.joinGroup(InetAddress.getByAddress("224.0.0.1"));
String message;
do
{
byte[] smsg=new byte[100];

DatagramPacket pack=new DatagramPacket(smsg,smsg.length);
mul.receive(pack);
message=new String(pack.getData());
System.out.println(pack.getAddress()+" : "+message);
}
while(!message.equals("close"));
mul.close();
}
catch(Exception e)
{
System.out.println(e);
}
}
}
}

```

### **Output:**

```
E:\EX9>javac msource.java
```

```
E:\EX9>java msource
```

```
Enter the text to send:
```

```
Jawaharlal Nehru Engineering College
```

```
E:\EX9>javac mulcli.java
```

```
E:\EX9>java mulcli
```

```
/10.1.50.10 : Jawaharlal Nehru Engineering College
```

### **Conclusion:**

Multicast is communication between a single sender and multiple receivers on a network.

### **Application:**

Video Conferencing

Computer Supported Common Work.

Distributed interactive simulation.

Large scale distributed (super)computing.

Distributed Games

## Lab Exercise: 10

Exercise No 5: ( 2 Hours) – 1 Practical

**AIM:** Implementation of Congestion Control Algorithm (Leaky Bucket Algorithm.)

**Objective:** Student should be able to develop a program on Leaky Bucket Algorithm.

### **Theory: LeakyBucket**

The congesting control algorithms are basically divided into two groups: open loop and closed loop. Open loop solutions attempt to solve the problem by good design, in essence, to make sure it does not occur in the first place. Once the system is up and running, midcourse corrections are not made. Open loop algorithms are further divided into ones that act at source versus ones that act at the destination.

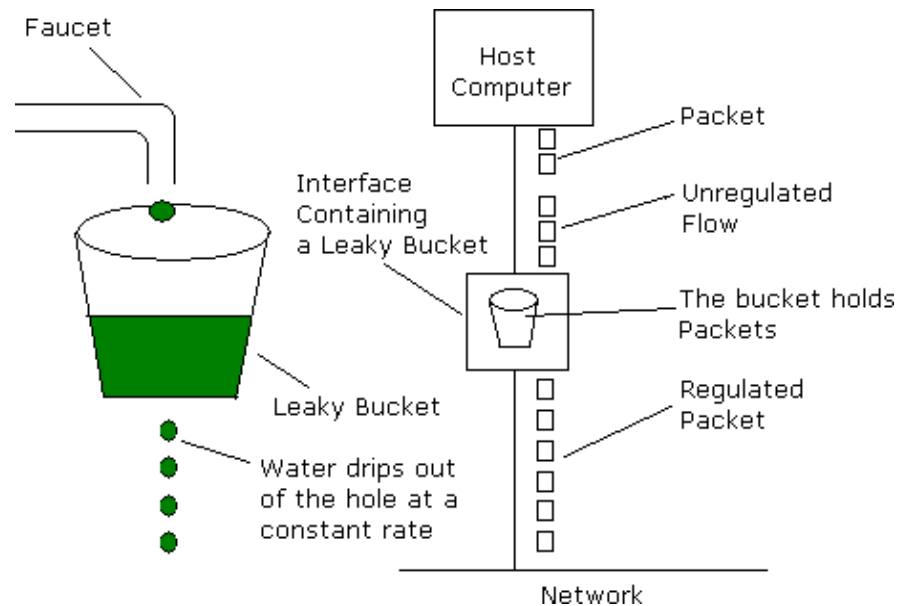
In contrast, closed loop solutions are based on the concept of a feedback loop if there is any congestion. Closed loop algorithms are also divided into two sub categories: explicit feedback and implicit feedback. In explicit feedback algorithms, packets are sent back from the point of congestion to warn the source. In implicit algorithm, the source deduces the existence of congestion by making local observation, such as the time needed for acknowledgment to come back.

The presence of congestion means that the load is (temporarily) greater than the resources (in part of the system) can handle. For subnets that use virtual circuits internally, these methods can be used at the network layer.

Another open loop method to help manage congestion is forcing the packet to be transmitted at a more predictable rate. This approach to congestion management is widely used in ATM networks and is called **traffic shaping**.

The other method is the leaky bucket algorithm. Each host is connected to the network by an interface containing a leaky bucket, that is, a finite internal queue. If a packet arrives at the queue when it is full, the packet is discarded. In other words, if one or more process are already queued, the new packet is unceremoniously discarded. This arrangement can be built into the hardware interface or simulated by the host operating system. In fact it is nothing other than a single server queuing system with constant service time.

The host is allowed to put one packet per clock tick onto the network. This mechanism turns an uneven flow of packet from the user process inside the host into an even flow of packet onto the network, smoothing out bursts and greatly reducing the chances of congestion.



### Algorithm:

#### Pseudo code:

```
void main()
{
    int i, packets[10], content=0, newcontent, time, clk, bcktsize, oprate;
    for(i=0; i<5; i++)
    {
        packets[i]=rand()%10;
        if(packets[i]==0) --i;
    }

    printf("\n Enter output rate of the bucket: \n");
    scanf("%d", &oprate);
    printf("\n Enter Bucketsize\n");
    scanf("%d", &bcktsize);
    for(i=0; i<5; ++i)
    {
        if((packets[i]+content)>bcktsize)
        {
            if(packets[i]>bcktsize)
```

```

printf("\n Incoming packet size %d greater than the size of the bucket\n",packets[i]); else
printf("\n bucket size exceeded\n");

}

else

{

newcontent=packets[i];
content+=newcontent;
printf("\n Incoming Packet : %d\n",newcontent);
printf("\n Transmission left : %d\n",content);
time=rand()%10;
printf("\n Next packet will come at %d\n",time);
for(clk=0;clk<time && content>0;++clk)
{

printf("\n Left time %d",(time-clk));
sleep(1);
if(content)

{

printf("\n Transmitted\n");
if(content<oprare) content=0;
else

content=content-oprate;

printf("\n Bytes remaining : %d\n",content);

}

else

printf("\n No packets to send\n");

}

}

}

}

```

### **Output:**

Enter output rate: 100

Packet no 0 Packet size = 3

Bucket output successful Last  
3 bytes sent

Packet no 1 Packet size = 33

Bucket output successful Last  
33 bytes sent

Packet no 2 Packet size = 117

Bucket output successful  
100 bytes outputted.  
Last 17 bytes sent

Packet no 3 Packet size = 95

Bucket output successful

### **Conclusion:**

The open loop congestion management is widely used in ATM networks and is called traffic shaping. The leaky bucket is an algorithm used in packet switched computer networks and telecommunications networks. It can be used to check that data transmissions, in the form of packets.

## Lab Exercise: 11

Exercise No 6: ( 2 Hours) – 1 Practical

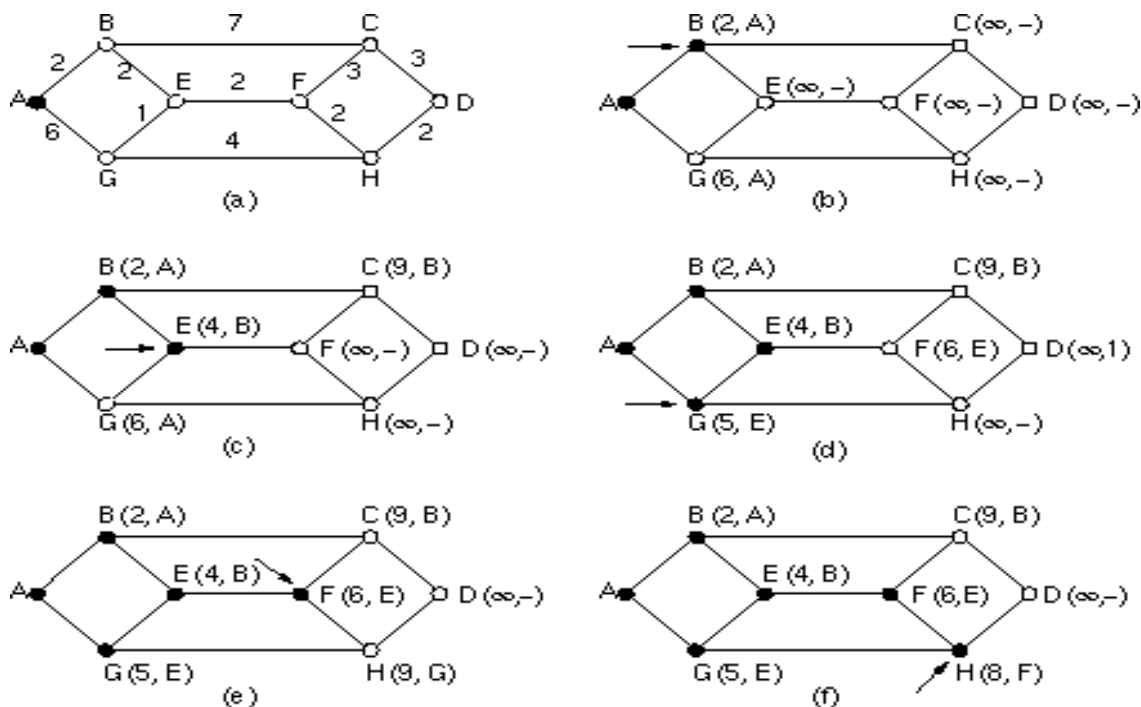
### AIM: Implementation of DIJKSTRA'S SHORTEST PATH ALGORITHM

**Objective:** Student should be able to develop a program on dijkstra's shortest path algorithm

#### Theory :

#### DIJKSTRA'S SHORTEST PATH ALGORITHM:

Dijkstra's algorithm is very similar to Prim's algorithm for minimum spanning tree. Like Prim's MST, we generate a *SPT (shortest path tree)* with given source as root. We maintain two sets, one set contains vertices included in shortest path tree, and other set includes vertices not yet included in shortest path tree. At every step of the algorithm, we find a vertex which is in the other set (set of not yet included) and has minimum distance from source.



**Shortest Path Algorithm** - Determines the optimal path through a graph based upon the criteria of number of hops, distance, cost, etc.

- Topology based, that is it determines connections to use from one router to another.



- Static, must be recomputed as routers added/deleted.
- Can be used to construct a sink tree with the destination the sink tree root. One possible application would be for each router to compute the sink tree of a network with itself as the root to determine the optimal output line incoming packets should be transmitted to reach their destination

The shortest path algorithm is in fact a general graph minimization algorithm where the metrics of the graph edges can represent distance, cost, time, etc. It is important in networking because of the desirability of minimizing network resources and is therefore employed by many routing algorithms.

The general idea behind the algorithm is to compute the *shortest path* from the *destination* node to other nodes until the *starting node* is reached. When complete each node points to its predecessor node on its own shortest path to the *destination*. From the *starting node*, the shortest path to the *destination* is then through the list of *predecessor* nodes.

Queue	Path	Closed
A		
B2, G6		A
E4, G6, C9	(B, A)	A, B
G5, G6, F6, C9	(E, B), (B, A)	A, B, E
F6, C9, H9	(G, E), (E, B), (B, A)	A, B, E, G
H8, C9, H9, C9	(F, E), (G, E), (E, B), (B, A)	A, B, E, G, F
C9, C9, D10	(H, F), (F, E), (G, E), (E, B), (B, A)	A, B, E, G, F, H
D10, D12	(C, F), (H, F), (F, E), (G, E), (E, B), (B, A)	A, B, E, G, F, H, C
	(D, H), (C, F), (H, F), (F, E), (G, E), (E, B), (B, A)	A, B, E, G, F, H, C, D

### Algorithm:

```
1 Initialization:
2 S = {u}
3 for all nodes v
4 if v adjacent to u
5   D(v) = c(u,v)
6 else D(v) =
7   _
8 Loop
9 find w not in S with the smallest D(w)
10 add w to S
11 update D(v) for all v adjacent to w and not in S:
12   D(v) = min {D(v), D(w) + c(w,v)}
13 until all nodes in S
```

### Output:

Vertex	Distance from Source
0	0
1	4
2	12
3	19
4	21
5	11
6	9
7	8
8	14

### Conclusion:

- 1) The code calculates shortest distance, but doesn't calculate the path information. We can create a parent array, update the parent array when distance is updated (like prim's implementation) and use it to show the shortest path from source to different vertices.
- 2) The code is for undirected graph, same dijkstra function can be used for directed graphs.

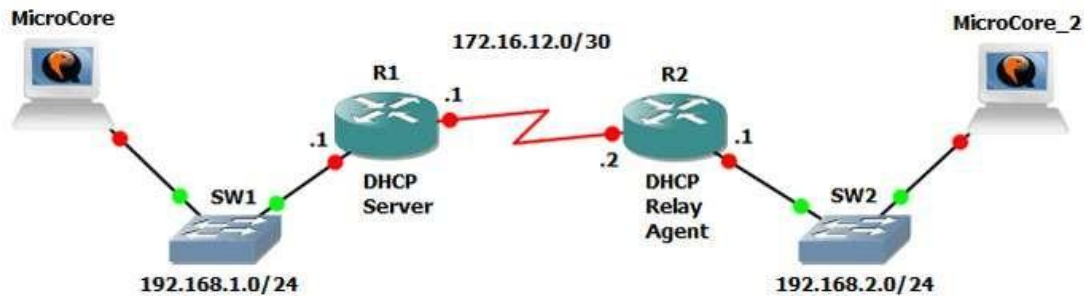
## Lab Exercise: 12

Exercise No 7: ( 2 Hours) – 1 Practical

**AIM: Simulation or implementation of DHCP.**

**Objective :** Student should be able dynamically assigning network addresses to hosts.

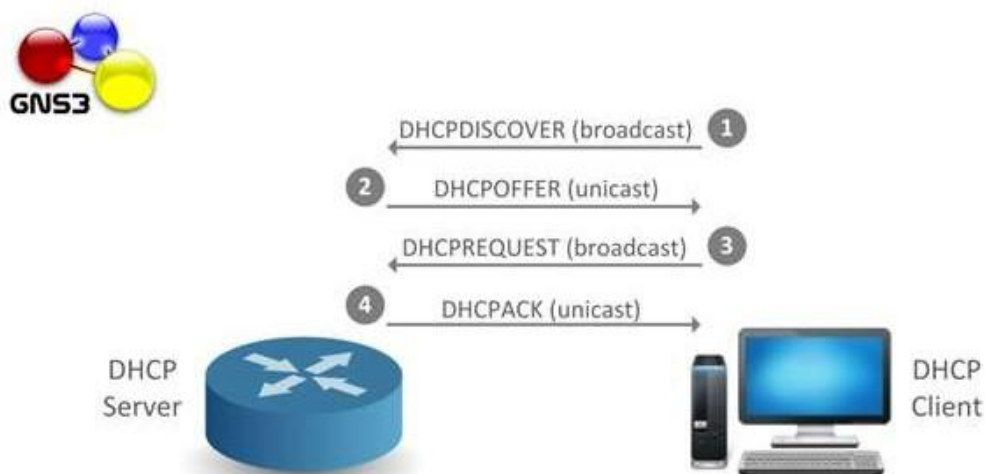
**Theory:**



DHCP (**dynamic host configuration protocol**) is used for dynamically assigning network addresses to hosts. DHCP operates on a client-server model, where a DHCP server sends configuration information to DHCP clients. The CCNA (Cisco Certified Network Associate) exams require you to understand DHCP and be able to configure and verify it on Cisco routers. In this article we will provide an introduction to DHCP, going into details relevant to CCNA. We will also develop a topology in GNS3 to practice the concepts hands on. The current article belongs to the GNS3 Labs for CCNA series and assumes you already have a GNS3 installation in working order

DHCP is based on BOOTP (bootstrap protocol), which provides the framework for passing configuration information to hosts on a TCP/IP network. The main advantage of DHCP over BOOTP is that you don't need to configure MAC addresses of all clients on the DHCP server.

The graphic below shows the message exchange between a DHCP server and client:



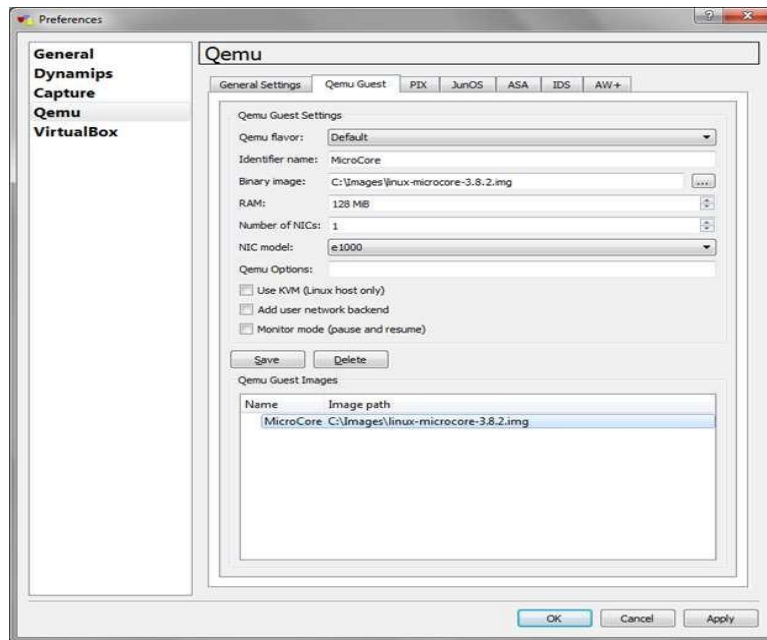
The steps shown in the graphic are explained below:

- The operating system on the host is configured to obtain network configuration via DHCP, so the host, acting as DHCP client, sends a DHCPDISCOVER broadcast message to locate a DHCP server.
- A DHCP server on the local subnet offers configuration parameters, including an IP address, to the client in a DHCPOFFER unicast message.
- The DHCP client returns a formal request for the offered IP address to the server in a DHCPREQUEST broadcast message.
- The DHCP server confirms that the IP address has actually been allocated for use by the client by returning the final DHCPACK unicast message.

The interaction of a DHCP server and client looks just like what we presented when the client and server are both on the same broadcast domain/subnet. If that's not the case, a DHCP relay agent is needed, in addition to client and server. A DHCP relay agent is any host that forwards DHCP packets between DHCP clients and DHCP servers. In practice, DHCP servers are often centrally located and you almost always need a DHCP relay agent to forward messages back and forth between the client and server. A DHCP relay agent does more than simply forwarding DHCP packets like a router forwarding IP packets. The relay agent receives DHCP messages and then generates a new DHCP message to send on another interface. The Cisco IOS (Internetwork Operating System) running on Cisco routers includes both DHCP server and relay agent software and we are going to cover the configuration of both features in this article.

### **DHCP Server Configuration and Verification:**

In the first article of the series (GNS3 Labs for CCNA: Getting Started), we set up GNS3 and created a simple topology with two routers, but we did not add any hosts. In this article, it will probably be useful to add a couple of hosts to see our DHCP server in action, actually assigning addresses. So let's learn how to add hosts to GNS3 before moving on to DHCP configuration.



The GNS3 0.8.6 all-in-one package that we are using comes bundled with version 0.11.0 of the Qemu emulator. You will use Qemu to emulate a lightweight distribution of Linux called Micro Core.

You should download the Qemu appliance for Linux Micro Core from <http://sourceforge.net/projects/gns-3/files/Qemu%20Appliances/linux-microcore-3.8.2.img> and save it to the directory C:\Images on your Windows system. You need also to configure GNS3 by going to Edit > Preferences > Qemu and then, on the tab labeled Qemu Guest, do as shown in the graphic below. Finally, you have to press the Save and OK buttons in succession.

You have to double click the downloaded **topology.net** file and GNS3 should launch automatically. You have to start all devices once the topology is loaded in GNS3. The devices will all come alive with initial configurations. First, you should go to R1 and add the following configuration to make it act as a DHCP server for directly connected hosts as well as for hosts connected to R2 on the remote LAN.

The command `show ip dhcp binding` reports IP addresses assigned and client MAC addresses associated with those addresses.

```
R1#show ip dhcp binding
```

```

IP address      Client-
ID/             Lease
Hardware

```

192.168.1.51	0100.abb6.8edc.00	Mar 02 2002 12:02 AM	Automatic
192.168.2.51	0100.abe3.5202.00	Mar 02 2002 12:30 AM	Automatic

The command **show ipdhcp pool** reports the number of addresses leased so far, along with some other details on current utilization of addresses in the DHCP pool.

Conclusion:

DHCP servers detect such conflicts by pinging the new IP address before assigning it to a client. The show ipdhcp conflict command lists all address conflicts known to the DHCP server. The DHCP server avoids assigning the addresses from the list to any clients until the list is manually cleared using the clear ipdhcp conflict command.

## Lab Exercise: 13

Exercise No 8: ( 2 Hours) – 1 Practical

**AIM: Simulation or Implementation of FTP.**

**Objective:** Student should be able to develop program on FTP and to promote sharing of files.

**Theory :**

✓ **FTP :**

- The FTP protocol is set up by using two connections instead of only one.
- In the model, the user-protocol interpreter (after you typed in ftp) initiates the control connection.
- The control connection follows the TELNET protocol.
- At the initiation of the user, standard FTP commands are generated by the user protocol interpreter and transmitted to the server process via the control connection.
- Standard replies are sent from the server protocol interpreter to the user protocol interpreter over the control connection in response to the commands.
- The FTP commands specify the parameters for the data connection and the nature of file system operation.
- The user-DTP or its designate should "listen" on the specified data port, and the server initiate the data connection and data transfer in accordance with the specified parameters.
- It should be noted that the data port need not be in the same host that initiates the FTP commands via the control connection, but the user or the user-FTP process must ensure a "listen" on the specified data port.
- It ought to also be noted that the data connection may be used for simultaneous sending and receiving.

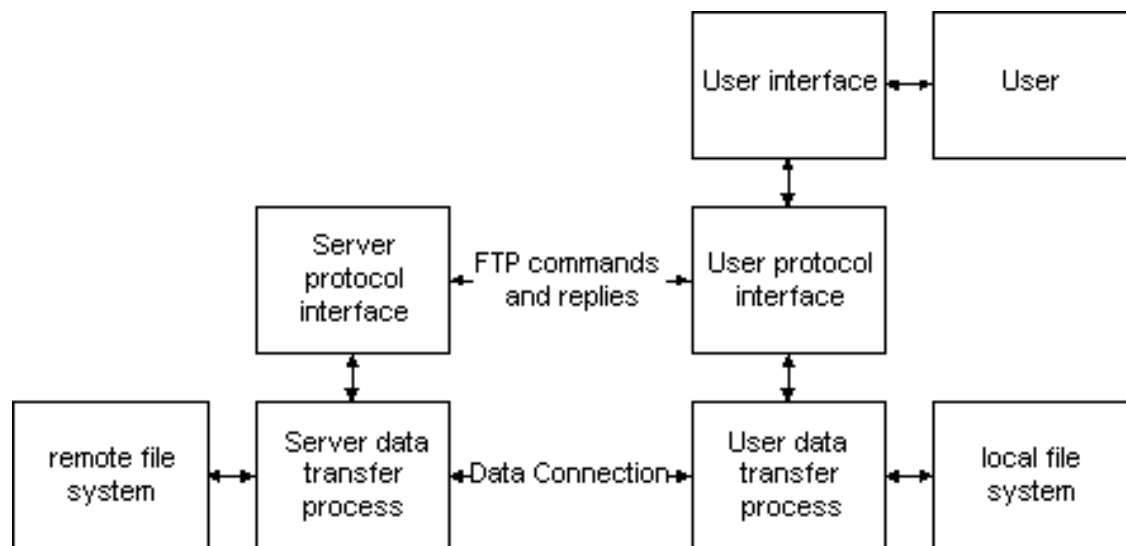


Figure:FTP

### Algorithm:

#### Server:

1. Include the necessary header files.
2. Create a socket using socket function with family AF\_INET, type as SOCK\_STREAM.
3. Initialize server address to 0 using the bzero function.
4. Assign the sin\_family to AF\_INET, sin\_addr to INADDR\_ANY, sin\_port to dynamically assigned port number.
5. Bind the local host address to socket using the bind function.
6. Listen on the socket for connection request from the client.
6. Accept connection request from the Client using accept function.
7. Within an infinite loop, receive the file name from the Client.
8. Open the file read the file contents to a buffer and send the buffer to the Client.

#### Client:

1. Include the necessary header files.
2. Create a socket using socket function with family AF\_INET, type as SOCK\_STREAM.
3. Initialize server address to 0 using the bzero function.
4. Assign the sin\_family to AF\_INET.
5. Get the server IP address and the Port number from the console.
6. Using gethostbyname function assign it to a hostent structure, and assign it to sin\_addr of the server address structure.
7. Within an infinite loop, send the name of the file to be viewed to the Server.
9. Receive the file contents, store it in a file and print it on the console.



## **Output:**

### **Server:**

```
(Host Name:Root1) [root@localhost  
4ita33]# vi ftps.c [root@localhost  
4ita33]# cc ftps.c [root@localhost  
4ita33]# ./a.out Server is Running...  
FILE REACHED  
File output :it is implementated...
```

### **Client:**

```
(Host Name:Root2) [root@localhost  
4ita33]# vi ftpc.c [root@localhost  
4ita33]# cc ftpc.c [root@localhost  
4ita33]# ./a.out Enter the filename:  
cse.txt  
Sending the file content Data  
sent.....
```

## **Conclusion:**

To implement FTP application, where the Client on establishing a connection with the Server sends the name of the file it wishes to access remotely. The Server then sends the contents of the file to the Client, where it is stored.



## Lab Exercise: 14

Exercise No 12: ( 2 Hours) – 1 Practical

**AIM: Implementation of chatting application using socket programming..**

**Objective:** Student should be able to develop a program on chatting application

**Theory:**

Inet Address

```
public class InetAddress
    extends Object implements
    Serializable
```

It is a class in Java which represents an Internet Protocol (IP) address. An instance of an InetAddress consists of an IP address and possibly corresponding hostname. The Class represents an Internet address as Two fields: 1- Host name (The String)=Contain the name of the Host.2- Address(an int)=The 32 bit IP address. These fields are not public, so we can't Access them directly. There is not public constructor in InetAddress class, but it has 3 static methods that returns suitably initialized InetAddress Objects namely Public String getHostName() , public byte[] getAddress() and public String getHostAddress()

**BufferedReader class:**

The BufferedReader class is used for fast reading operations of texts from a characterinput stream. It can be used to read single characters, arrays, and lines of data. The size of buffer may or may not be specified. The readLine() method of the BufferedReader class can be used to get the next line of characters from a file, and the skip(long n) method can be used to skip n number of characters.

**Client-Server Model:**

The client–server model of computing is a distributed application that partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients. Often clients and servers communicate over a computer network on separate hardware, but both client and server may reside in the same system. A server machine is a host that is running one or more *server programs which share their resources with clients*. A client does not share any of its resources, but *requests a server's content or service function*. Clients therefore initiate communication sessions with servers which *await incoming requests*

**Socket Programming:**

A *socket* is one of the most fundamental technologies of computer networking. Sockets allow applications to communicate using standard mechanisms built into network hardware and operating systems. Although network software may seem to be a relatively new “Web” phenomenon, socket technology actually has been employed for roughly two decades.

A socket represents a single connection between exactly two pieces of software. More than two pieces of software can communicate in *client/server* or *distributed* systems (for example, many Web browsers can simultaneously communicate with a single Web server) but multiple sockets are required to do this. Sockets are *bidirectional*, meaning that either side of the connection is capable of both sending and receiving data. Libraries implementing sockets for Internet Protocol use TCP for streams, UDP for datagrams, and IP itself for raw sockets

**Algorithm:**

```
import java.io.*; import
java.net.*; class
chatclient
{
public static Socket soc;
public static void main(String args[])throws IOException
{
try
{
InetAddress a=InetAddress.getLocalHost();
soc=new Socket(a,0202);
BufferedReader d=new BufferedReader(new InputStreamReader(System.in)); BufferedReader
in=new BufferedReader(new InputStreamReader(soc.getInputStream())); PrintWriter out=new
PrintWriter(new BufferedWriter(new BufferedWriter(new
OutputStreamWriter(soc.getOutputStream()))),true);
String s;
s=in.readLine();
System.out.println(s);
s=in.readLine();
System.out.println(s);
s=d.readLine();
while(true)
{
out.println(s);
s=in.readLine();
System.out.println("Server:> "+s);
if(s.equalsIgnoreCase("Chat is closing"))
break;
System.out.println("Message : ");
s=d.readLine();
if(s.equalsIgnoreCase("end")) break;
}
}
finally
{
soc.close();
}
}
}
```

```
import java.io.*; import
java.net.*; class
chatserver
{
public static void main(String args[])throws IOException
{
ServerSocket s=new ServerSocket(0202); try
{
while(true)
{
Socket soc=s.accept(); try
{
new chat(soc);
}
catch(IOException e)
{
soc.close();
}
}
}
finally
{
s.close();
}
}
```

**Output:**

```
E:\EX8>javac chatserver.java
```

```
E:\EX8>java  
chatserverMessage :  
Hi Client, how are you? BE
```

```
CSE:> I am fine.
```

```
E:\EX8>javac chatclient.java
```

```
E:\EX8>java chatclient
```

```
Chat sessions begins..Server :
```

```
Your name please :
```

```
BE CSE  
Server:> Hi Client, how are you?
```

```
Message :
```

```
I am fine.
```

**Conclusion:**

Sockets allow applications to communicate using standard mechanisms built into network hardware and operating systems.

## **Lab Exercise: 15**

Exercise No 11: ( 2 Hours) – 1 Practical

**AIM: Analysis of enterprise Network Monitor Tool such as Wireshark.**

**Objective:** Student should be able to develop

1. Using Wireshark to monitor the LAN.
2. Browse and Analyze packets from a live network or from a previously saved capture file.

**Theory :**

**Wireshark:**

Wireshark is comprised of three main windows, or panes.

The top pane is the packet list pane. It displays a summary of each packet captured.

The middle pane is the packet details pane. It displays the packet selected in the top pane in more detail.

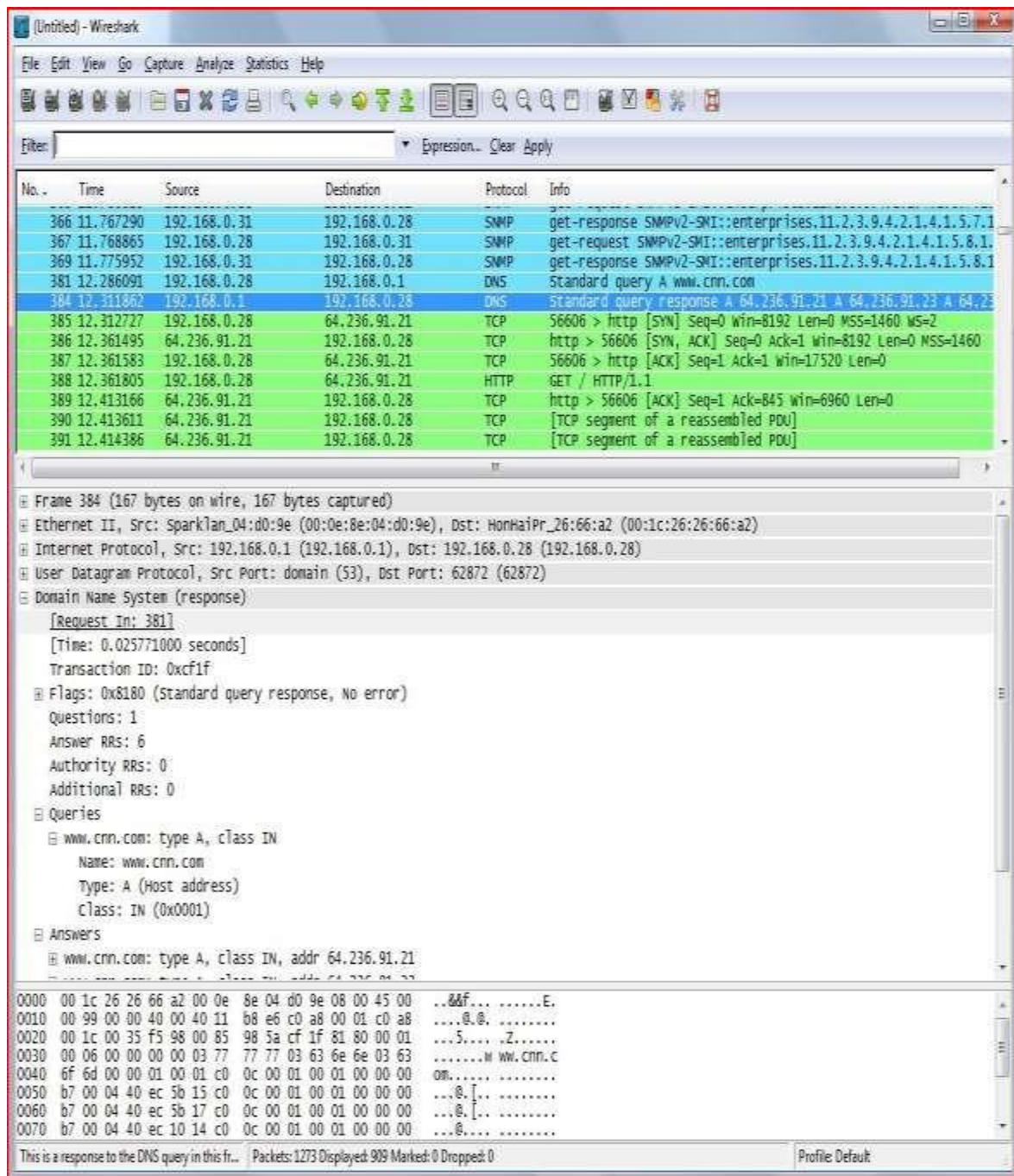
The bottom pane is the packet bytes pane. It displays the data from the packet selected in the top pane, and highlights the field selected in the tree view pane.

In addition to the three main panes, there are four elements of interest on the bottom of the Wireshark main window.

Menu items: a drop down menu of all operations

Main Toolbar: shortcut icons to main operations by Wireshark





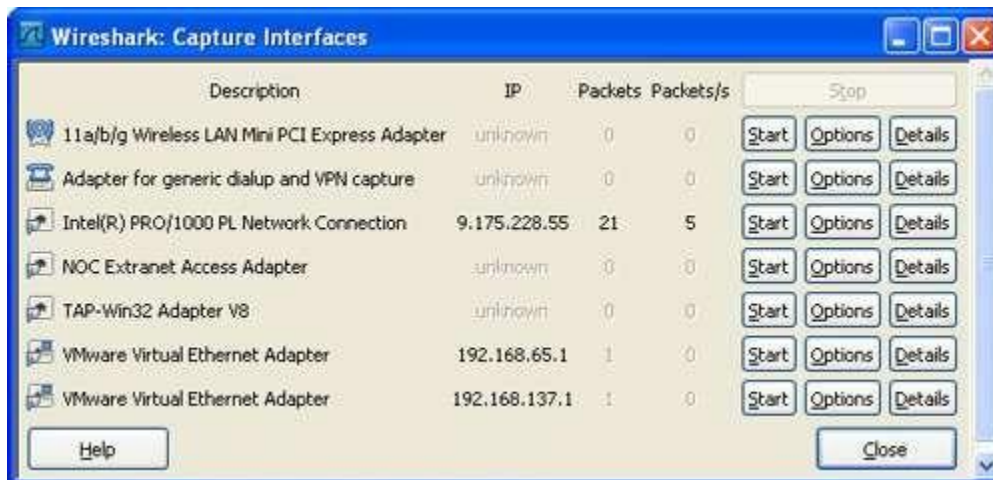
**Filter Toolbar:** for filter construction

The leftmost button labeled "Filter:" can be clicked to bring up the filter construction dialog.

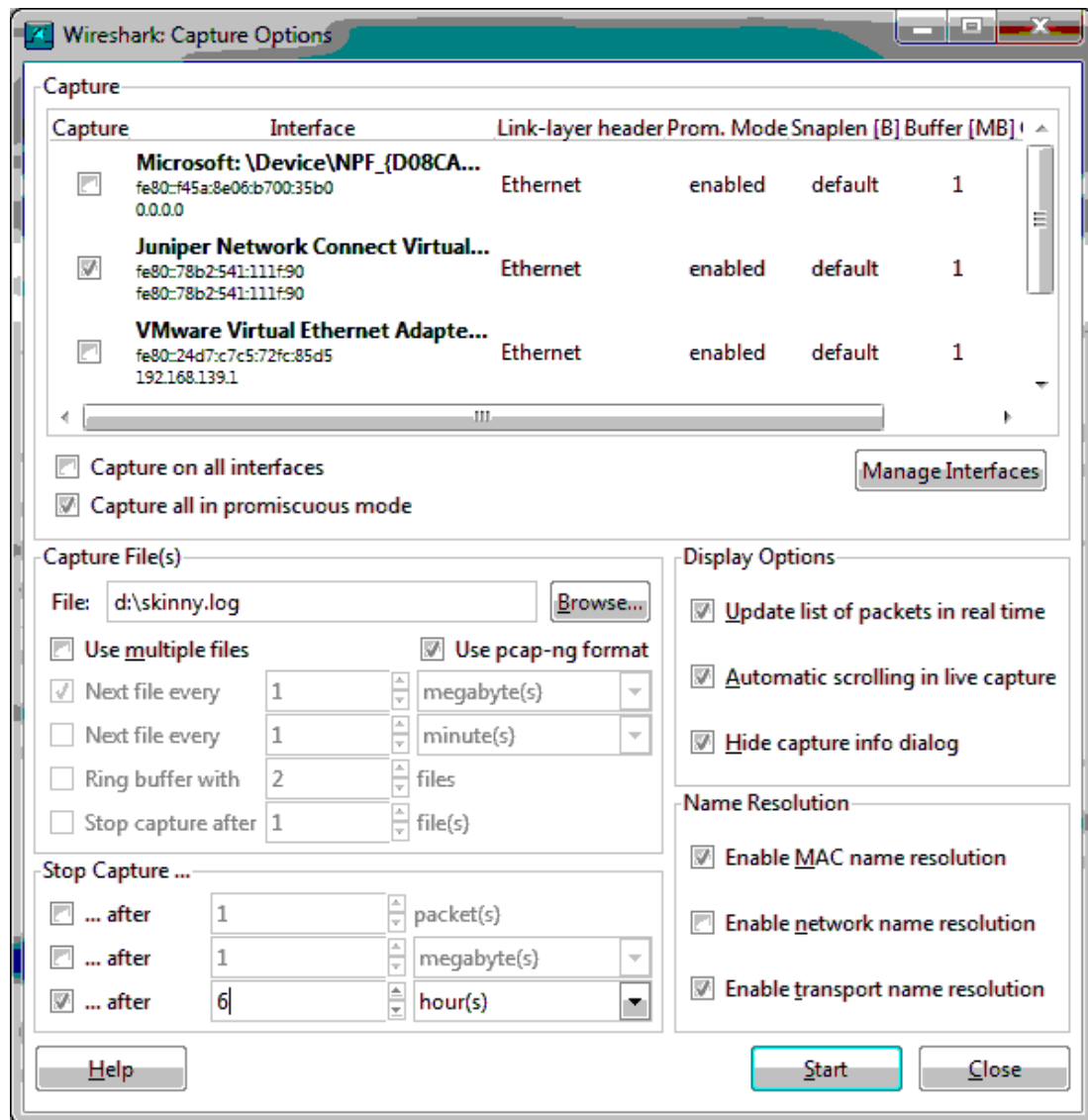
The middle text box provides an area to enter or edit filter strings. This is also where the current filter in effect is displayed. The right middle button labeled "Clear" clears the current filter.

**Status Toolbar:** it gives helpful hints about things like Capturing packets in Wireshark

The Capture Preferences dialog box Before you start capturing, the capture interface must be specified using capture ->interfaces to bring up a dialog box similar to the one shown below



You can select Start to start capturing from a specific interface or you may click on Options for the specified interface to bring a dialog box like the one below. This is the Capture Preferences dialog box:



You may arrive immediately at this dialog box from the menu item **capture ->options**. You can also choose the interface you want to capture on and the filter options plus many other options as shown. The options are explained as:

### Filtering Options:

**-Capture packets in promiscuous mode:** Usually a network card will only capture the traffic to its own network address. If you want to capture all traffic that the network card can “see”

**-Limit each packet to xy bytes:** Will limit the maximum size to be captured of each packet, this includes the link-layer header and all subsequent headers. This can be useful when an error is known to be in the first 20 bytes of a packet, for example, as the size of the resulting capture file will be reduced.

**-Capture Filter:** Use a capture filter to reduce the amount of packets to be captured (will be explained later) Storing options

**-File:** You can choose the file to which captured data will be written. If you don't enter something here a temporary file will be used.

**-Use multiple files:** Instead of using a single capture file, multiple files will be created. The generated filenames will contain an incrementing number and the start time of the capture. For example, if you choose "/foo.cap" in the "File" field, files like "/foo\_00001\_20040205110102.cap", "/foo\_00002\_20040205110102.cap", will be created. This feature can be useful if you do long term capturing, as working with a single capture file of several GB usually isn't very fast.

Stop condition options

These three fields should be obvious; the capture process will be automatically stopped if one of the selected conditions is exceeded. Display while capturing options

**-Update list of packets in real time:** Using this will show the captured packets immediately on the main screen. Don't change the default preferences. Start capturing. Keep capturing for 2-3 minutes during which try to activate the different protocols by invoking different applications (e.g. internet browsing, streaming, email, ASU online, ftp, etc)

## Capture Filter Syntax

The following is a short description of the capture filter language syntax. For a further reference, have a look at: [http://www.tcpdump.org/tcpdump\\_man.html](http://www.tcpdump.org/tcpdump_man.html)

A capture filter takes the form of a series of primitive expressions, connected by conjunctions (and/or) and optionally preceded by not:

[not] primitive [and|or [not] primitive ...] A

primitive is simply one of the following:

[src|dst] host <host>

ether [src|dst] host <ehost>

gateway host <host>

[src|dst] net <net> [{mask <mask>}] {len<len>}]

[tcp|udp] [src|dst] port <port>

less|greater<length>ip|ether

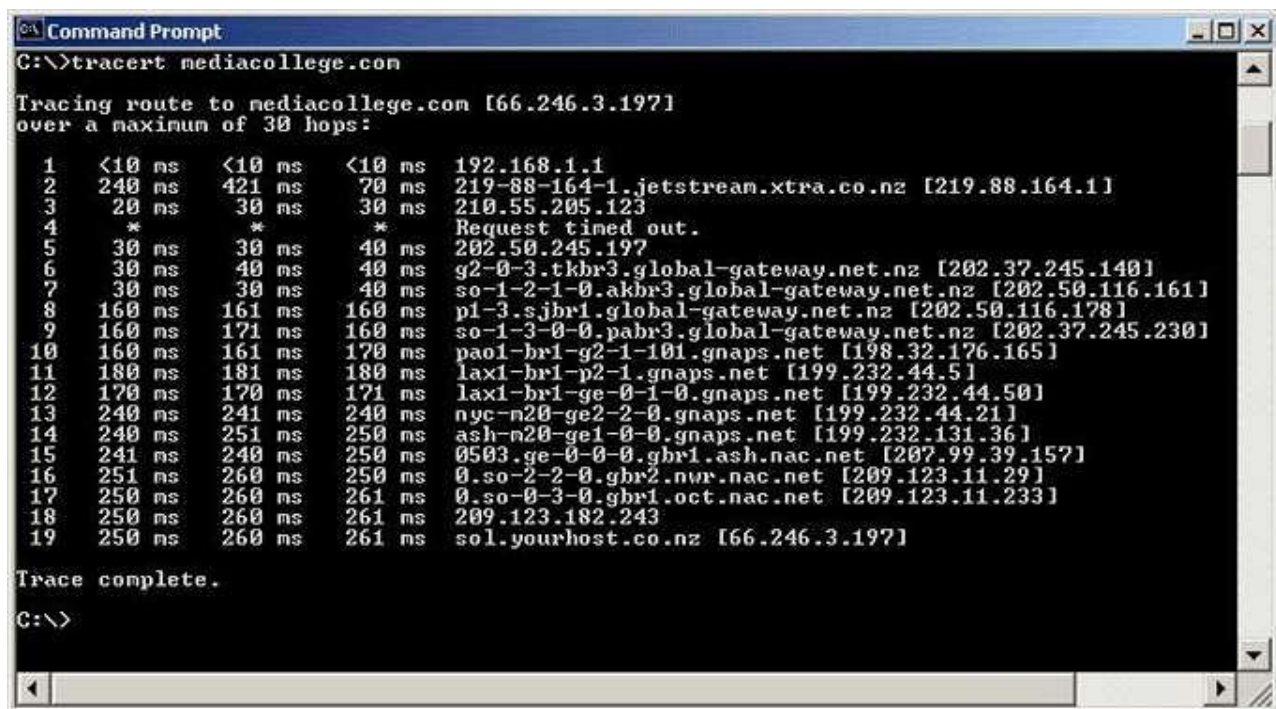
proto

<protocol>ether|ipbroadcast|mul

ticast

<expr>relop<expr>

## Output:



```
Command Prompt
C:\>tracert mediacollege.com

Tracing route to mediacollege.com [66.246.3.197]
over a maximum of 30 hops:

  1  <10 ms  <10 ms  <10 ms  192.168.1.1
  2  240 ms  421 ms  70 ms  219-88-164-1.jetstream.xtra.co.nz [219.88.164.1]
  3  20 ms  30 ms  30 ms  210.55.205.123
  4  * * * Request timed out.
  5  30 ms  30 ms  40 ms  202.50.245.197
  6  30 ms  40 ms  40 ms  g2-0-3.tkbr3.global-gateway.net.nz [202.37.245.140]
  7  30 ms  30 ms  40 ms  so-1-2-1-0.akbr3.global-gateway.net.nz [202.50.116.161]
  8  160 ms  161 ms  160 ms  pl-3.sjbr1.global-gateway.net.nz [202.50.116.178]
  9  160 ms  171 ms  160 ms  so-1-3-0-0.pabr3.global-gateway.net.nz [202.37.245.230]
 10  160 ms  161 ms  170 ms  paol-br1-g2-1-101.gnaps.net [198.32.176.165]
 11  180 ms  181 ms  180 ms  lax1-br1-p2-1.gnaps.net [199.232.44.5]
 12  170 ms  170 ms  171 ms  lax1-br1-ge-0-1-0.gnaps.net [199.232.44.50]
 13  240 ms  241 ms  240 ms  nyc-n20-ge2-2-0.gnaps.net [199.232.44.21]
 14  240 ms  251 ms  250 ms  ash-n20-ge1-0-0.gnaps.net [199.232.131.36]
 15  241 ms  240 ms  250 ms  0503.ge-0-0-0.gbr1.ash.nac.net [207.99.39.157]
 16  251 ms  260 ms  250 ms  0.so-2-2-0.gbr2.nwr.nac.net [209.123.11.29]
 17  250 ms  260 ms  261 ms  0.so-0-3-0.gbr1.oct.nac.net [209.123.11.233]
 18  250 ms  260 ms  261 ms  209.123.182.243
 19  250 ms  260 ms  261 ms  sol.yourhost.co.nz [66.246.3.197]

Trace complete.
C:\>
```

## Conclusion:

Wireshark, a network analysis tool formerly known as Ethereal, captures packets in real time and display them in human-readable format. Wireshark includes filters, color-coding and other features that let you dig deep into network traffic and inspect individual packets.

