**EXPERIMENT -1**

**AIM:** PREPARE A CASE STUDY ON HARDWARE COMPONENTS USED IN COMPUTER NETWORKS LABORATORY.

***THEORY:***

Computer network components are the *major parts* which are needed to *install the software*. Some important network components are **switch**, **cable**, **hub**, **router**, and **modem**. Depending on the type of network that we need to install, some network components can also be removed. For example, the wireless network does not require a cable.

Following are the hardware components:

***REPEATER***

A **repeater** is an electronic device that receives a signal and retransmits it. Repeaters are used to extend transmissions so that the signal can cover longer distances or be received on the other side of an obstruction. Some types of repeaters broadcast an identical signal, but alter its method of transmission. When an information-bearing signal passes through a communication channel, it is progressively degraded due to loss of power. For example, when a telephone call passes through a wire telephone line, some of the power in the electric current which represents the audio signal is dissipated as heat in the resistance of the copper wire. The longer the wire is, the more power is lost, and the smaller the amplitude of the signal at the far end. So, with a long enough wire the call will not be audible at the other end. Similarly, the farther from a radio station a receiver is, the weaker the radio signal, and the poorer the reception. A repeater is an electronic device in a communication channel that increases the power of a signal and retransmits it, allowing it to travel further. Since it amplifies the signal, it requires a source of electric power.

***MODEM***

Another significant network component is modern. The term Modem is the shortened version of the name modulator-demodulator. Modern provides two-way communication facility between a computer network and telephone network. As Wide Area Network uses the existing telephone network to connect to a distant network, it always uses a modern to dial-up the telephone network.

The modem converts the digital data from the computer into useful analog signals that can transmit through a telephone network. Similarly, signals from the telephone channels are converted back into digital data s1suitable for a computer.

***HUB***

The network hub is a centralised distribution point for all data transmission in a network. Hub may also refer to as a concentrator Data packet from a NIC arrives at the hub. The hub receives and rebroadcasts them to other computers connected to it. In general, the hub network is a passive device. It does not know the destination of a received data packet. Hence, it is required to send copies to all the hub connections. Stackable hubs are hubs that can be stacked or interconnected to make a single hub appearance.

They are useful for vendors to make hubs of a size suitable to customer requirement.

Non-stackable hubs cannot be interconnected. They always provided only a fixed number of connections.

The hubs that connect to the network backbone are known as active hubs. The hubs, which connect only to active hubs, are known as passive hubs.

Intelligent hubs contain a special firmware that can be accessed by remote workstations. The firmware is known as the Simple Network Management Protocol (SNMP). Network performance and Network status data read from SNMP.

***ROUTER***

Routers guide and direct network data, using packets that contain various kinds of data—such as files, communications, and simple transmissions like web interactions.

The data packets have several layers, or sections, one of which carries identifying information such as sender, data type, size, and most importantly, the destination IP (Internet protocol) address. The router reads this layer, prioritizes the data, and chooses the best route to use for each transmission. A router also can act as a bridge. Such a router is known as a router. The router receives the packet and examines whether it supports the protocol used by the packet. If not, it only drops the packet. The packet is bridged using the physical address information.

***SWITCH***

A switch is a device in a computer network that connects other devices together. Multiple data cables are plugged into a switch to enable communication between different networked devices. Switches manage the flow of data across a network by transmitting a received network packet only to the one or more devices for which the packet is intended. Each networked device connected to a switch can be identified by its network address, allowing the switch to direct the flow of traffic maximizing the security and efficiency of the network.

Role in network:

Switches are most commonly used as the network connection point for hosts at the edge of a network. In the hierarchical internetworking model and similar network architectures, switches are also used deeper in the network to provide connections between the switches at the edge.

In switches intended for commercial use, built-in or modular interfaces make it possible to connect different types of networks, including Ethernet, Fibre Channel, RapidIO, ATM, ITU-T G.hn and 802.11. This connectivity can be at any of the layers mentioned. While the layer-2 functionality is adequate for bandwidth-shifting within one technology, interconnecting technologies such as Ethernet and token ring is performed more easily at layer 3 or via routing. Devices that interconnect at the layer 3 are traditionally called routers.

***GATE WAY***

Two different networks can be connected using a gateway. For example, a mainframe can be connected and accessible to a PC network using a gateway. Unlike routers, a gateway converts the format of the data sent between two networks. A router adds only addressing information to the data packet. Routers never change the content of the message. However, a gateway has to identify the protocols used in the networks, and recognise the data format and convert the message format into a suitable format to be accepted by the other network. Wide area networks often use gateways because there is a large number of different networks present in a WAN. Gateways provide excellent connectivity to different kinds of networks on the Internet.

***BRIDGE***

A bridge in a computer network is one kind of network device, used to separate a network into sections. Every section in the network represents a collision domain that has separate bandwidth. So that network performance can be improved using a bridge. In the OSI model, a bridge works at layer-2 namely the data link layer. The main function of this is to examine the incoming traffic and examine whether to filter it or forward it.

The working principle of a bridge is, it blocks or forwards the data depending on the destination MAC address and this address is written into every data frame. In a computer network, a bridge separates a LAN into different segments like segment1 & segment2, etc and the MAC address of all the PCs can be stored into the table. For instance, PC1 transmits the data to PC2, where the data will transmit to the bridge first. So the bridge reads the MAC address & decides whether to transmit the data to segment1 or segment2. Therefore, the PC2 is accessible in segment1, which means the bridge transmits the data in segment1 only & eliminates all the connected PCs in segment2. In this way, the bridge reduces traffic in a computer network.

**EXPERIMENT-2**

**AIM:** PREPARE A CASE STUDY ON OSI MODEL.

**THEORY:**

Open Systems Interconnection (OSI is a reference tool for understanding data communications between any two networked systems. It divides the communications processes into seven layers. Each layer both performs specific functions to support the layers above it and offers services to the layers below it. The three lowest layers focus on passing traffic through the network to an end system. In the OSI reference model, the communications between a computing system are split into seven different abstraction layers: Physical, Data Link, Network, Transport, Session, Presentation, and Application.

## Physical Layer

## The physical layer defines the means of transmitting raw bits rather than logical data packets over a physical link connecting network nodes. The bit stream may be grouped into code words or symbols and converted to a physical signal that is transmitted over a lathe and milling transmission medium. The physical layer provides an electrical, mechanical, and procedural interface to the transmission medium. The shapes and properties of the electrical connectors, the frequencies to broadcast on, the modulation scheme to use and similar low-level parameters, are specified here. Within the semantics of the OSI network architecture, the physical layer translates logical communications requests from the data link layer into hardware-specific operations to affect transmission or reception of electronic signals.

## Data Link Layer

## The Data Link Layer is the second layer in the OSI (open systems interconnection) seven-layer reference model. The data link layer has a number of specific functions to carry out.

## The three main functions of the Data Link layer are:

## 1. Providing a well-defined service interface to the network layer.

## 2. Dealing with transmission errors.

## 3. Regulating the flow of data so that slow receivers are not swamped by fast senders.

## Network Layer

The Network Layer is the layer that provides data routing paths for network communication. Data is transferred in the form of packets via network logical paths in an ordered format controlled by the network layer. The network layer is considered as the backbone of OSI model. It selects and manages the best logical path for data transfer between nodes. This layer contains hardware devices such as routers, bridges, firewalls and switches but it actually creates logical images of the most efficient communication route and implements it with a physical medium. Network layer protocols exist in every host or router. The router examines the header fields of all the IP packets that pass through it.

## Transport Layer

This layer provides transparent transfer of data between end systems, or host and is responsible for end – to – end error recovery and flow control. It ensures complete data transfer. The transporter layer responsible as it offers end – to – end communication between end devices through a network. Depending on the application, the transport layer either offer reliable, connection – oriented or connectionless, best effort communications.

## Session Layer

The session layer controls the conversations between different computers. A session or connection between machines is set up, managed, and determined at layer 5. Session layer services also include authentication and reconnections.

## Presentation Layer

The presentation layer formats or translates data for the application layer based on the syntax or semantics that the application accepts. Because of this, it at times also called the syntax layer. This layer can also handle the encryption and decryption required by the application layer.

## Application Layer

The application layer is to provide protocols for exchange of information between application processes and provided all services directly comprehensible to application programs. The application layer identifies communication partners, resource availability, and synchronizes communication. Transferring of files disturbing the results to the user is also done in this layer. Mail services, directory services, network resource etc are services provided by application layer.

**EXPERIMENT -3**

**Aim**: Implement the Bellman Ford Algorithm used for designing distance vector routing protocol.

**Theory:** 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 rout remains in force for an entire user session (e.g., login session at a terminal or a file).

**ALGO:**

1. Structure node

Distance array, Reach array

Array of Objects of structure Node :re

1. Input number of node: n and graph/dist array [n][n]

Re[i].ds[j]=dist[i][j] and re[i].re[j]=j

1. Do loop start

For loop i (0 -> n)

For loop j (0-> n)

For loop k (0 -> n)

if ((re[i].ds[j]) > (re[i].ds[k] + re[k].ds[j]))

re[i].ds[j] = re[i].ds[k] + re[k].ds[j]

re[i].reach[j] = k

flag = 1

For loop k ends

For loop j ends

For loop i ends

While(flag)

1. Print reach and ds for every Re object.

**Code:**

#include<bits/stdc++.h>

Using namespace std;

struct node {

int ds[20];

int reach[20];

} re[10];

int main()

{

int dis[20][20], n;

cout << "Number of nodes : " << endl;

cin >> n; cout << "Distance matrix : " << endl;

for (int i = 0; i < n; i++) {

for (int j = 0; j < n; j++) {

# cin >> dis[i][j];

# dis[i][i] = 0;

# re[i].ds[j] = dis[i][j];

# re[i].reach[j] = j;

# }

# }

# int flag;

# do {

# flag = 0;

# for (int i = 0; i < n; i++) {

# for (int j = 0; j < n; j++) {

# for (int k = 0; k < n; k++) {

# if ((re[i].ds[j]) > (re[i].ds[k] + re[k].ds[j])) {

# re[i].ds[j] = re[i].ds[k] + re[k].ds[j];

# re[i].reach[j] = k;

# flag = 1;

# }

# }

# }

# }

# } while (flag);

# for (int i = 0; i < n; i++) {

# cout << "Router info : " << i + 1 << endl;

# cout << "Dest\t Next Hop\t Distance" << endl;

# for (int j = 0; j < n; j++)

# printf("%d\t\t%d\t\t%d\n", j+1, re[i].reach[j]+1, re[i].ds[j]);

# }

# return 0;

# }

# **Result:**

# 

**EXPERIMENT – 4**

**AIM:** IMPLEMENT THE DIJKSTRA ALGORITHM USED FOR LINK STATE ROUTING.

**THEORY:** Link state routing is a technique in which each router shares the knowledge of its neighbourhood with every other router in the internetwork. Each node uses Dijkstra's algorithm on the graph to calculate the optimal routes to all nodes. The Link state routing algorithm is also known as Dijkstra's algorithm which is used to find the shortest path from one node to every other node in the network. The Dijkstra's algorithm is an iterative, and it has the property that after kth iteration of the algorithm, the least cost paths are well known for k destination nodes.

**ALGO:**

1. Create parent, ggset, distance array of size (no. of routers)
2. Initialise parent array with -1
3. Initialise distance array with INT\_MAX
4. Initialise ggset with false
5. For loop start (0 -> n-1)

For loop start(0 -> n)

If(ggset = false && distance[v] <= min)

Minimum distance (u) =v;

For loop End

For loop v start(0 -> n)

If(graph[u][v]+distance[u]<distance[v])

Distance = distance[u] + graph[u][v]

Parent[v]=u

For loop u end

For loop end

1. Print distance array and path using parent array.

**Code:**

#include<bits/stdc++.h>

using namespace std;

int len;

int mindis(int distance[], bool ggset[])

{

int min = INT\_MAX, min\_index;

for (int v = 0; v < len; v++)

if (ggset[v] == false && distance[v] <= min)

min = distance[v], min\_index = v;

return min\_index;

}

void print(int parent[], int j)

{

if (parent[j] == - 1)

return;

print(parent, parent[j]);

cout<<" "<<j;

}

int main()

{

ios\_base:: sync\_with\_stdio(false);

cin.tie(NULL); cout.tie(NULL);

cin>>len;

vector<vector<int>> graph(len,vector<int>(len,0));

for(int i=0;i<len;i++){

for(int j=0;j<len;j++){

cin>>graph[i][j];

}

}

int src=0;

int distance[len];

bool ggset[len];

int parent[len];

for (int i = 0; i < len; i++)

{

parent[0] = -1;

distance[i] = INT\_MAX;

ggset[i] = false;

}

distance[src] = 0;

for (int count = 0; count < len - 1; count++)

{

int u = mindis(distance, ggset);

ggset[u] = true;

for (int v = 0; v < len; v++)

if (!ggset[v] && graph[u][v] &&

distance[u] + graph[u][v] < distance[v])

{

parent[v] = u;

distance[v] = distance[u] + graph[u][v];

}

}

int st = 0;

cout<<"Vertex\t Distance\t\tPath\n";

for (int i = 1; i < len; i++)

{

cout<<st<<" -> "<<i<<"\t\t"<< distance[i]<<"\t\t\t"<< st;

print(parent, i);

cout<<"\n";

}

return 0;

}

***Result:***

****