CNS Practical - 6

Title : Implement Link State / Distance Vector Routing Protocol

Aim : Write a program to implement link state / Distance Routing Protocol to find suitable path for transmissio

Objective : To understand and implement Link State or 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. or . . 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 circits internally, routing decisions are made only when a new established route is being set up. The latter case is sometimes called for an entire user session.

Routing algorithms can be grouped into two major classes : adaptive and non-adaptive. 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 is compute in advance, offline & downloaded to routers when the network is booted. its called static routi

*. Two algorithms in particular, distance vector routing and link state routing are the most popular. Distance vector routing maintain a table giving the best know distance to each destination and which line to get there.

These tables are updated by exchanging info with neighbour's. The distance vector routing algo is sometime called by distributed Bellman-Ford routing algorithm ~~is sometimes called by other names~~. and Ford-Fulkerson algorithm.

In distance vector routing each router maintains a routing table indexed by & containing one entry for, each router in subnet.
This entry contains two parts: preferred outgoing line to use for that destination & an estimate of the time or distance to that destination.

The router is assumed to know the 'distance' to each of its neighbour. If the metric is hops the distance is just one hop. If the metric is queue length, router simples examines each queue. If 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.

Algorithm :

1. Start
2. By convention the distance of node itself is assigned to zero & when a node is unreachable the distance is accepted as 999.
3. Accept the input distance matrix from the user (dm[][]) that represents the distance between each node in the network.
4. Store the distance between nodes in a suitable var
5. Calculate the minimum distance between 2 nodes by iterating
   - If distance between two nodes is larger than the calculated alternate available path, replace the existing distance with the calculated distance
6. Print the shortest path calculated
7. Stop.

Conclusion: Thus we implemented State / Distance vector routing algorithm to find suitable path for transmission.

A.I.S.S.M.S
INSTITUTE OF INFORMATION TECHNOL
Kennedy Road, Near R.T.O., Pune - 411

CNS Practical 7

Title : Configuring of 3 router network.

Aim : Use packet tracer tool for configuration of 3 router network using one of the following protocol RIP / OSPF / BGP

Objective : To use packet tracer tool to configure 3 router network.
Understand different protocols RIP | OSPF | BGP

Theory :

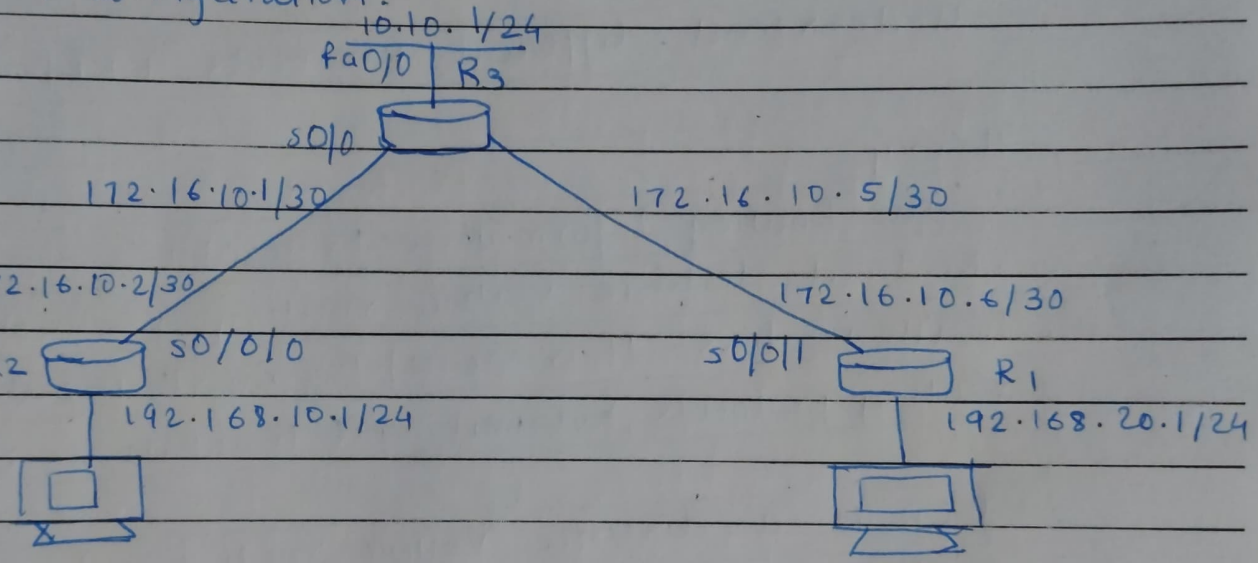
The Routing Information Protocol is an industry standard distance vector routing protocol available to all vendors. It is one of the most popular dynamic routing protocols, however, its limitations.

- RIP is a distance vector routing protocol.
- It has a default administrative distance of 120
- RIP can go maximum hop of 15
- It uses hop counts to determine the shortest path to a remote network.
- RIP has no support for classless addressing & variable length subnet mask (VLSM) RIP V2 does.

Features of RIP:

1. Updates of the network are exchanged periodically.
2. Updates (routing information) are always broadcast
3. Full routing tables are sent in updates.
4. Routers always trust routing information received from neighbour routers. This is also known as Routing on rumours.

RIP Configuration:



Consider the above-given topology which has 3-routers R1, R2, R3.

+ R1 has IP address 172.16.10.6/30 on s0/0/1, 192.168.20.1/24 on fa0/0.
+ R2 has IP address 172.16.10.2/30 on s0/0/0, 192.168.10.2/30 on s0/0/0, fa0/0.
+ R3 has IP address 172.16.10.5/30 on s0/1, 172.16.10.1/30 on s0/0, 10.10.10.1/24 on fa0/0

Configure RIP for R1 :

R1(config) # router rip
R1 (config-router) # network
192.168.20.0
R1 (config-router) # network
172.16.10.4
R1 (config-router) # version 2
R1 (config-router) # no auto-summary

Configure RIP for R2

R2 (config) # router rip
R2 (config-router) # network 192.168.10.0
R2 (config-router) # network 17.16.10.0
R2 (config-router) # version 2
R2 (config-router) # no auto-summary

Configure RIP for R3

R3 (config) # router rip
R3 (config-router) # network 10.10.10.0
R3 (config-router) # network 172.16.10.4
R3 (config-router) # version 2
R3 (config-router) # no auto-summary

* Conclusion : Thus we implement 3 router network configuration using RIP protocol.

CNS Practical 8

Title: TCP socket for wired network

Aim: Write a program using TCP socket for wired network for following

a. Say Hello to Each other
b. File Transfer
c. Calculator

Objective: Use & Study programs, TCP socket for wired networks to implement certain tasks mentioned.

Theory:

+ **Socket Programming:**

The Berkeley socket interface, an API, allows communication between hosts or between processes on one computer, using the concept of a socket. It can work with many different I/O devices & drives although support for these depends on the OS implementation. This interface implementation is implicit for TCP/IP. & it is therefore one of the fundamental technologies underlying the Internet. Programmers can make the socket interfaces accessible at three different levels, most powerfully & fundamentally at RAW socket level.

* TCP : TCP provides the concept of a connection. A process creates a TCP socket by calling the sockets function with the parameters PF_INET or PF_INET6 and SOCK_STREAM

* Server : Setting up a simple TCP server involves the foll steps:

Creating a TCP socket with a call to socket() Binding socket to the listen port, with a call to bind(). Before calling bind(), a programmer must declare a sockaddr_in structure, clear it (with bzero() or memset()), & the sin_family & fill its sin_port fields. Converting short int to network byte order can be done by calling function htons().

Preparing socket to listen for connections, with a call to listen(). Accepting incoming connections via a call to accept().

* Client : Setting up a TCP client involves foll steps:

1. Creating a TCP socket, with a call to socket()
2. Connecting to server, passing sockaddr_in with sin_family set to AF_INET or AF_INET6, sin_port set to port endpoint & sin_addr set to IPV4 or IPv6 address of listening server.
3. Communicating with server by send()ing & recv()ing. Close by using close().

*. Algorithm:

Server Program

1. Open the server socket:

ServerSocket server = new ServerSocket(PORT);

2. Wait for Client Request:

Socket client = server.accept());

3. Create I/O streams for communicating to client

4. Perform communication with client

Receive from client: String line = is.readLine();

Send to client: os writeBytes("Hello\n")

5. Close socket:

client.close();

* Conclusion: Thus we have successfully implemented the socket programming for TCP using C.

CNS Practical 9

Title : UDP sockets to enable file transfer

Aim : Write a program using UDP sockets to enable file transfer (Script, Text, Audio & Video one) file each) between two machines.

Objective :

Implement file transfer using UDP sockets

Theory :

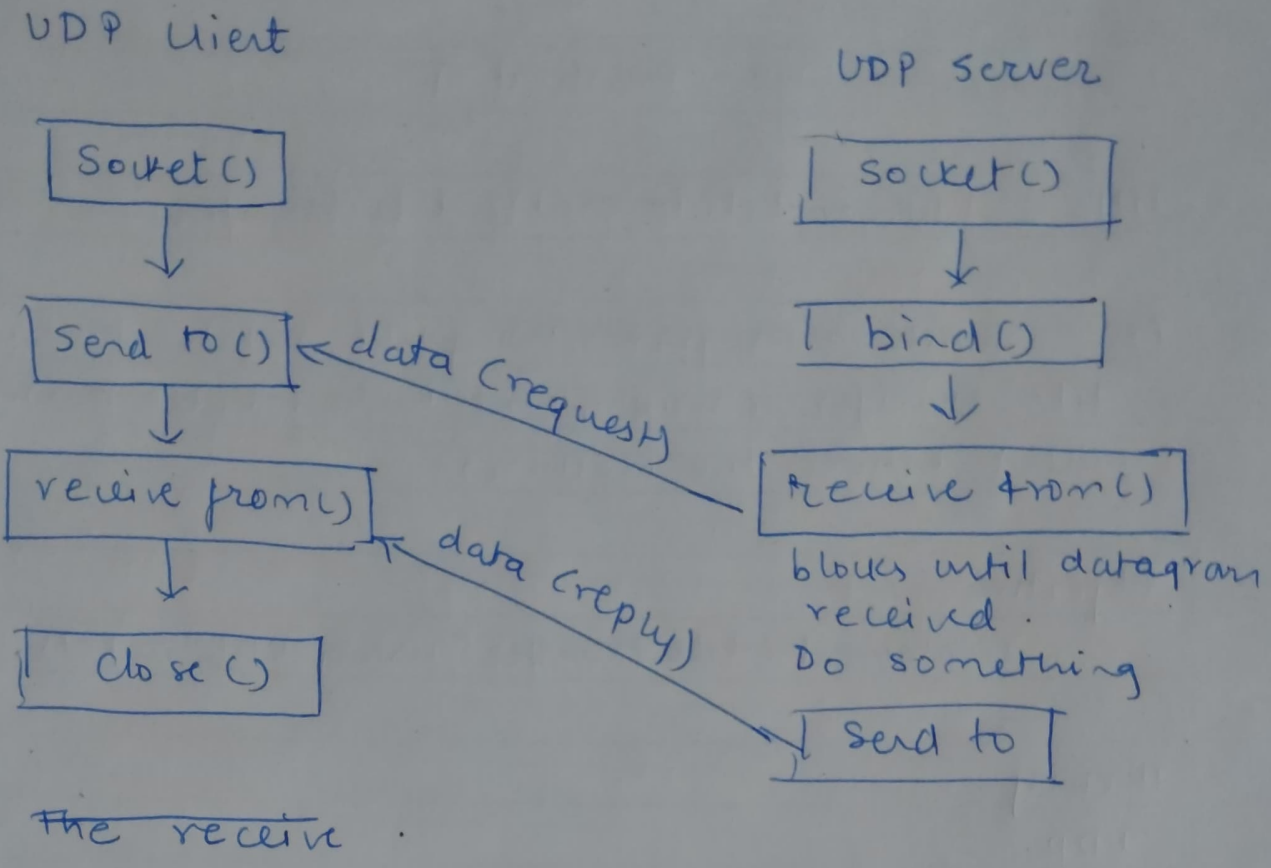
UDP -

It consists of a connectionless protocol with no gurantee of delivery. UDP packets may arrive out of order, become duplicated and arrive more than one or not arrive more than once or not arrive more than once or not arrive at all.

The space for UDP port numbers is a completely disjoint from that of TCP ports

There are a few steps involved in using sockets:

1] Create Socket
2] Identify Socket
3] Wait on server for the message
4] On the client send message
5] Send response back to client
6] Close socket


UDP Client
UDP Server
Socket()
Send to()
receive from()
Close()
socket()
bind()
receive from()
blocks until datagram received.
Do something
Send to
data (request)
data (reply)

The receive .

The receive from () function -
Similar to read() function with 3 additional arguments

Syntax -
Size_t receive from (int socket, void buffer, size-t n bytes, int flags, struct sockaddr * from socket address)

Send to() function
this function is very similar to send() function but 3 additional arguments are required.

* Algorithm :
Client -
1] Create a socket using socket() function
2] Send and receive data by means of receive from() and send to() function.

Server -
1] Create a socket with socket() function.
2] Bind the socket to an address using bind() function.
3] Send and receive data by means of recursion() & send to()

Conclusion : Here we have studied UDP programming to enable file transfer using two machines.

CNS Practical 10

Title: DNS Lookup

Problem Statement: Write a program for DNS lookup. Given an IP input it should return URL and vice versa.

Objective: Implement DNS lookup to return URL and IP.

Theory:

DNS: A DNS server is a computer server that contains a database of public IP address & their associated host names & in most cases servers to resolve or translate, those common names to IP addresses as requested DNS servers run special software & communicate with each other using special protocols.

The DNS domain name space is based on the concept of tree of named domains. Each level of tree represent either a branch or a leaf. A branch is a level where name is used to identify a collection of named resources. A leaf represents a single name used once to indicate a specific resource

Why do we have DNS servers :-

When you ever "google.com" you only have to remember the URL.

Similarly for computers it is easier to remember IP address.

Therefor, DNS servers are required to translate domain name to IP address & load internet resources.

DNS Lookup -

It is the process by which a DNS record is returned from a DNS server.

Forward DNS lookup is using internet domain name to find IP address. Reverse DNS lookup is using an internet IP address to find a domain name forward DNS lookup is more common.

Algorithm -

1] Give input website address

2] Using Java/Python, extract IP address

3] Show internet address as name/address

4] Return IP address corresponding to domain name.

Application :

DNS server is a translator between the hoost name & the IP address.

* Conclusion - Hence we have studied the process of DNS lookup & have written a program for the same

CNS Practical 11

Title : Install & configure DHCP server

Problem Statement : Installing and configure DHCP server and write a program to install the software on remote machine.

Objective : Install & configure DHCP serve
Install software on remote machine.

Theory :

The Dynamic Host configuration Protocol (DHCP) is a standardized network protocol used on Internet Protocol (IP) network for dynmaically distributing network configuration parameters, IP networks for dy such as IP addresses for interfaces & services. With DHCP computers request IP addresses & networking parameters automatically from a DHCP server, reducing the need for a network administer or a user to configure these settings manually.

DHCP is a network service that enables host computers to be automatically assigned settings from a server as opposed to manually configuring each network host. Computers configured to be DHCP clients have no control over the settings they receive from the DHCP server, & the configuration is transparent to computer's user.

The most common settings provided by a DHCP server to DHCP clients include:

IP address & netmask
IP address of the default gateway to use
IP addresses of the DNS servers to use

However, a DHCP server can also supply configuration properties such as:

Host name
Domain name
Time server
Print server

A DHCP server can provide configuration settings using the following methods:

1. Manual allocation (MAC address)
- The method entails using DHCP to identify the unique hardware address of each network card connected to the network and then continually supplying a constant configuration each time the DHCP client makes a request to the DHCP server using that network device.

2. Dynamic Allocation (address pool)
- In this method, the DHCP server will assign an IP address from a pool of addresses (sometimes also called a range or scope) for a period of time or lease that is configured on server or until the client informs server that it doesn't need the addr.

3. Automatic allocation

* Using this method, the DHCP automatically assigns an IP address permanently to device selecting it from a pool of available addresses. Usually DHCP is used to assign a temporary address to a client but a DHCP server can allow an infinite lease time.

* Installation

At a terminal prompt enter the following command to install dhcpd;

```
sudo apt-get install isc-dhcp-server
```

We need to change the default configuration by editing /etc/dhcp/dhcpd.conf to suit our needs and particular configuration.

We also may need to edit /etc/default/isc-dhcp-serv to specify the interfaces dhcpd should listen to.

* Configuration:

Assign IP address randomly:-

```
#minimal sample /etc/dhcp/dhcpd.conf
default-lease-time 600;
max-lease-time 7200;
#subnet 192.168.1.0 netmask 255.255.255.0 {
range 192.168.1.150 192.168.1.200;
option routers 192.168.1.254;
option domain-name-servers 192.168.1.1, 192.168.1.2
option domain-name "srttc"; }
```

Conclusion: Thus we installed & configured DHCP server and installed software on remote machine.