

LAB - I - OUTCOME.

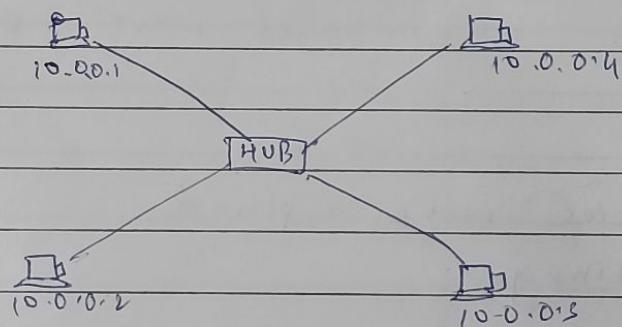
- In the first lab we learnt the important terminology related to the field of computer networks and visited the 'my first PT lab'.
We used the Cisco packet tracer to build our first network containing of personal computers.
- We also learnt how to configure the end devices and set IP addresses, rename them.
- Connecting devices and using the PDU tool to understand the network was also done.
- simple test messages were sent in realtime mode
- Scenarios were explored.
 - editing
 - creating new
 - deleting

AIM:-

LAB - 2 outcome

understanding simulation mode

- understanding the connections through the ping command.
- Creating a network with multiple end devices and a hub.
- testing the network with multiple configurations
- sending a packet, a simple PDU message from one end device to another in the network connected.



Observing the broadcast

Topo

9/11/2020

Practical
Lab

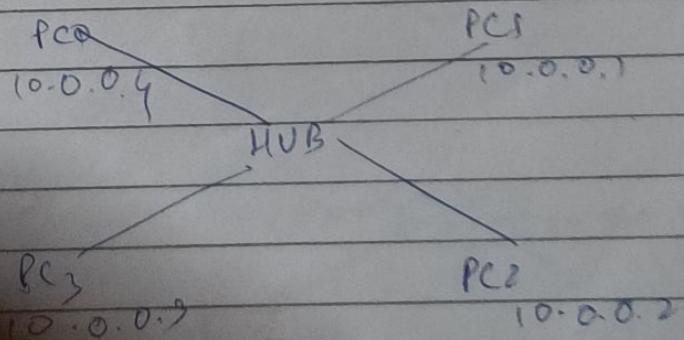
AIM:-

Simulate sending a simple PDU from source to a destination.

Procedure :-

- Create a network containing 4 PCs, one HUB and connect them to a single HUB through a copper straight through wire.
- Configure the device IP addresses on the same network and test it by pinging the connected device's IP using Ping command in the command prompt of PC0.
- Enter simulation mode and send a simple PDU message from one device to another and observe it being broadcasted when clicked on ^{auto} capture/play.
- Edit the filters of visible events to only observe necessary events.

Apology



Result :-

Successfully sent a PDU message from a specified source to a specified destination.

Meelima
17/11/2022

PAGE NO.:
DATE:

LAB - 3

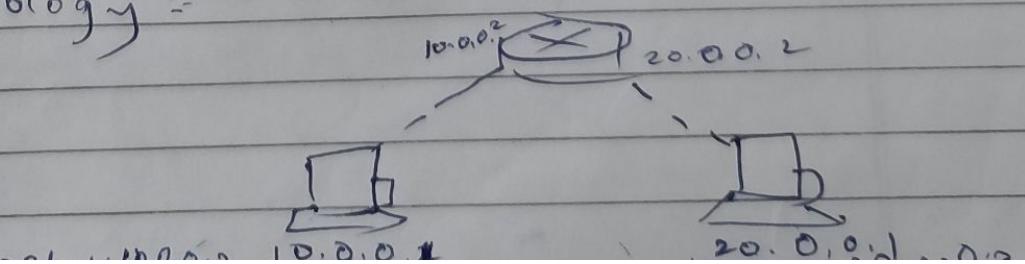
PAGE NO.:
DATE:

AIM:- To configure IP address to router in P.T. Explore ping response, destination unreachable, request timed out, Reply.

PROCEDURE:-

- Create a network containing two end devices and one router.
- Configure the IP address of the 2 end devices as a part of 2 different networks
- Configure the router using the CLI window using telnet command
 - enable
 - config terminal
 - interface Fa0/0
 - ip address 10.0.0.2 255.0.0.0
 - no shutdown
 - exit
 - do this with IP 20.0.0.8 as well
 - try to ping 10.0.0.2,
20.0.0.2,
20.0.0.1
 - configure gateways of end devices
 - try to ping again.

Topology:-



Result:-

During first ping set:-
ping to 10.0.0.2 from 10.0.0.1
was successful

but ping from 10.x network
to 20.x network was timed
out.

But after configuring the
gateway all set of ping were
successful.

Pings:- Before gateway

10.0.0.1 → 10.0.0.2

successful

10.0.0.1 → 20.0.0.2

request timed out

10.0.0.1 → 20.0.0.1

request timed out

20.0.0.1 → 20.0.0.2

successful

20.0.0.1 → 10.0.0.2

request timed out

20.0.0.1 → 10.0.0.1

request timed out

After gateway

$10 \cdot 0 \cdot 0 \cdot 1 \rightarrow 20 \cdot 0 \cdot 0 \cdot 1$
 $(10 \cdot 0 \cdot 0 \cdot 1 \rightarrow 20 \cdot 0 \cdot 0 \cdot 2)$
 $(10 \cdot 0 \cdot 0 \cdot 1 \rightarrow 10 \cdot 0 \cdot 0 \cdot 2)$
} successful

$20 \cdot 0 \cdot 0 \cdot 1 \rightarrow 10 \cdot 0 \cdot 0 \cdot 1$
 $20 \cdot 0 \cdot 0 \cdot 1 \rightarrow 10 \cdot 0 \cdot 0 \cdot 2$
 $20 \cdot 0 \cdot 0 \cdot 1 \rightarrow 20 \cdot 0 \cdot 0 \cdot 2$
} successful

seen : BEFORE GATEWAY

PC0 > ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:-

Reply from 10.0.0.2 : bytes = 32 time = 20ms

Reply from 10.0.0.2 : bytes = 32; time = 0ms TTL = 255
TTL = 35

Reply from 10.0.0.2 : bytes = 32; time = 0ms TTL = 255

Reply from 10.0.0.2 : bytes = 32; time = 0ms TTL = 255

Ping statistics for 10.0.0.2

Packets : sent = 4 ; received = 4

Lost = 0 (0% loss)

Approximate round trip times in

milli -seconds : Minimum = 0ms,

Maximum = 0ms, Average = 0 ms

Before gateway,

PC> ping 20.0.0.2

Pinging 20.0.0.2 with 32 bytes of
Data

Request timed out

request timed out

Request timed out

Request timed out.

Ping stats for 20.0.0.2:

packets : Sent = 4, Received = 0, Lost = 4
(100% loss)

AFTER GATEWAY

of 20.0.0.2 in 10.0.0.1

Reply

PC> ping 20.0.0.2

Pinging 20.0.0.2 with with 32

bytes

of data:

Reply from 20.0.0.2: bytes = 32, time = 0ms

Reply from 20.0.0.2: bytes = 32, time = 0ms
TTL = 255

Reply from 20.0.0.2: bytes = 32, time = 0ms
TTL = 255

Reply from 20.0.0.2: bytes = 32, time = 0ms

TTL = 255

Ping statistics for 20.0.0.2

Sent = 4 received = 4, Lost = 0

Minimum = 0°ms , Max = 0°ns , Average = 0°

9/11/22

L A B - 4

PAGE NO:
DATE:

AIM: to configure a static route through routers

PROCEDURE:

create a network with 2 end devices & 3 networks such that :

end devices are connected to routers through fast ethernet.

routers are serially connected to each other.

set IP of PC1 to 10.0.0.1

and gateway to 10.0.0.2 in config

set IP of PC2 to 40.0.0.1 and and gateway to 40.0.0.2

configure the routers' IP in the CLI window using commands

- enable
- config terminal
- ip interface fa0/0 OR se1/0
- ip address [IP] [SUBNET MASK]
- no shutdown
- exit

Do this for all the 3 routers such that it matches the topology drawn later.

Go to CLI of R1 and type

"show ip route"

For R1, 10.0.0.0 & 20.0.0.0

will be the only 2 networks connected

Go into config terminal & type

> ip route 30.0.0.0 255.0.0.0 20.0.0.2
 network subnet hop

Do this for all networks which are each connected to the router.

If we do not do this, we get

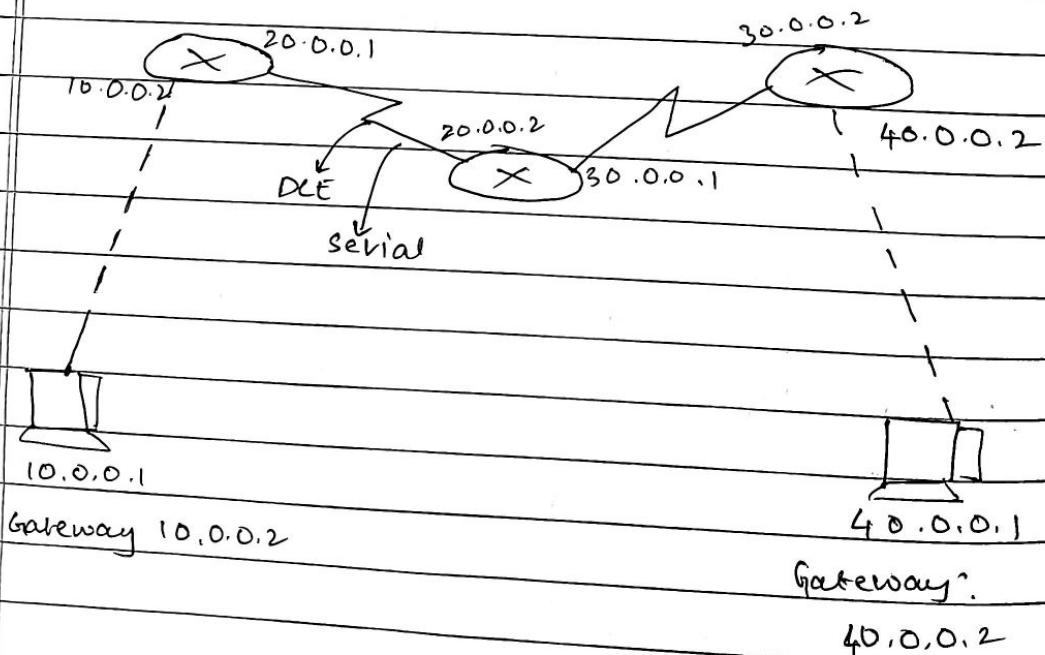
"Request timed out"

"Destination Host unreachable" errors

Request is timed out when router does not know if IP specified is connected.

Host is unreachable when router is NOT CONNECTED to that network.

TOPOLOGY:



RESULT

Before

get "de"

we try

not

before

"Request

a con

g does

After

In co

pc> ping

Reply

Reply br

Reply br

Reply G

Ping st

Pa

Needless
4/1/2022 APP

type
 20.0.0.2
 ↘
 hop

which are

ev.

get

"errors"

ip

router is

work.

X
 ↗
 40.0.0.2

gateway?

40.0.0.2

RESULT :-

Before configuration of routers, we get "Destination host unreachable" error if we try to ping a network that has not been specified in the IP route.

Before configuration of routers, we get "Request Timed out" if we try to ping a connected network that the router does not have an address for.

After proper configuration.

In command prompt of PC1 (10.0.0.1)

PC> ping 40.0.0.1

Reply from 40.0.0.1 : bytes = 32 time = 14 ms
 TTL = 25

Reply from 40.0.0.1 : bytes = 32 time = 2 ms TTL = 125

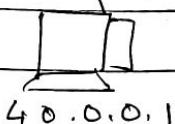
Reply from 40.0.0.1 : bytes = 32 time = 15 ms TTL = 125

Reply from 40.0.0.1 : bytes = 32 time = 5 ms TTL = 125

Ping statistics for 40.0.0.1 :

Packet(s) : sent = 4 Received = 4 Lost = 0
 (0% loss)

Needless to say Approximate round trip times in ms:
 Minimum = 2 ms, Maximum = 15 ms,
 Average = 9 ms



01/12/22

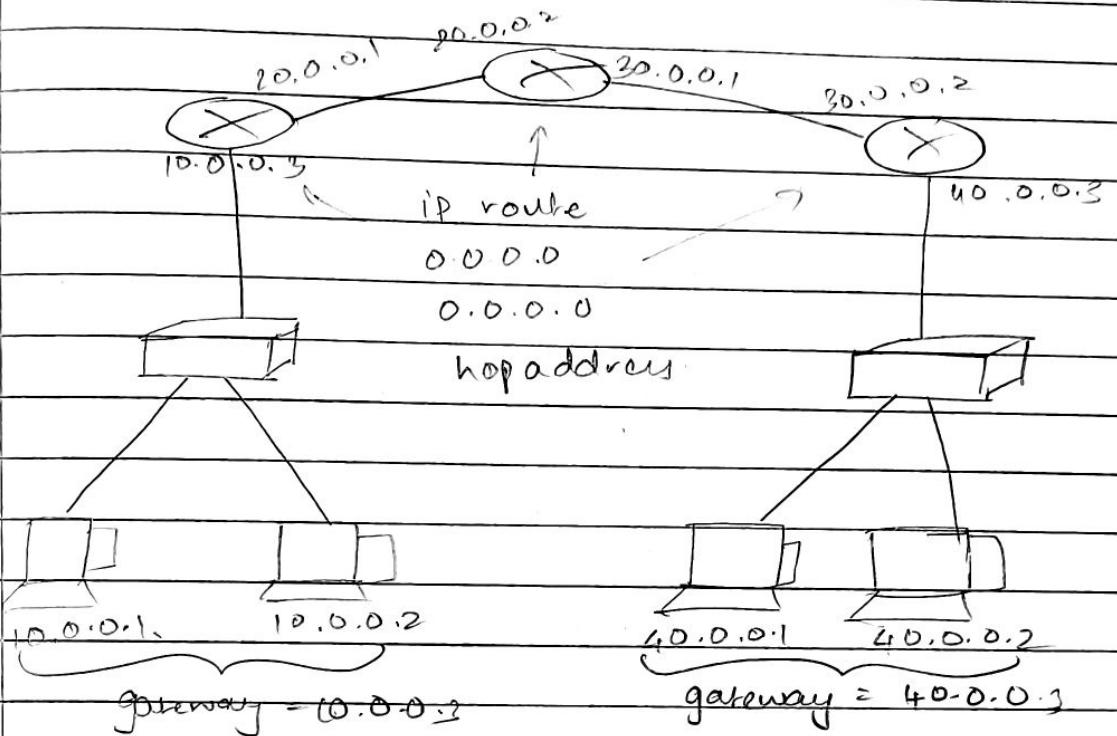
LAB - 5

AIM:-

PAGE NO:
DATE:

To configure default routes to routers

Topology :-



Procedure :

create a network following the topology above having 4 end devices, 3 routers & 2 switches.

configure the ip addresses of the end devices. \Rightarrow 10.0.0.1, 10.0.0.2 for the 2 connected to first network

& 40.0.0.1 & 40.0.0.2 for the other two.

configure the routers following the topology in the CLI windows.

set gateways for the PCs respectively

For each router, configure the default gateway by the command:

ip route 0.0.0.0 0.0.0.0 <hop add>
any any
1p SUBNET

* test by pinging all different combinations:-

Observations:-

① Before default route:

10.0.0.0 $\xrightarrow{\text{ping}}$ 20.0.0.1 100% successful

10.0.0.0 $\xrightarrow{\text{ping}}$ 20.0.0.2 Request Timed out

10.0.0.0 $\xrightarrow{\text{ping}}$ 30.0.0.1 unreachable

10.0.0.0 $\xrightarrow{\text{ping}}$ 40.0.0.1 unreachable

② After default route:

10.0.0.0 $\xrightarrow{\text{ping}}$ 20.0.0.1 SUCCESSFUL 100%

10.0.0.0 $\xrightarrow{\text{ping}}$ 20.0.0.2 50% successful

50% request timed out

10.0.0.0 $\xrightarrow{\text{ping}}$ 30.0.0.0 50% successful
50% timed out

10.0.0.0 $\xrightarrow{\text{ping}}$ 40.0.0.1 100% SUCCESSFUL.

0 <hop add>

JET

different

00% successful

request timed out

reachable

reachable

SUCCESSFUL 100%.

successful

request timed
out

successful

timed out

SUCCESSFUL.

Result:-

Default route was configured successfully

Neelima
1/2/2022

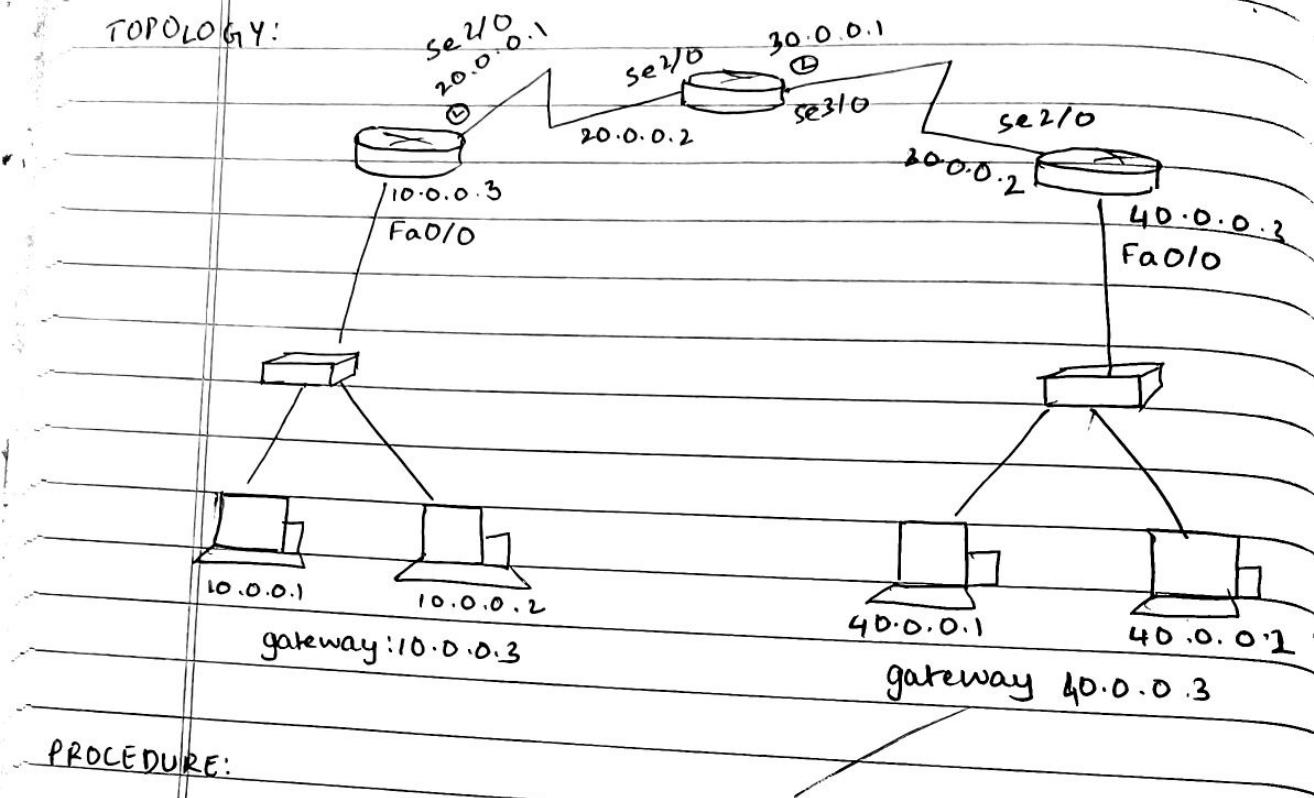
LAB - 6

8/12/12

PAGE NO.:
DATE:

AIM: To configure Routing Information Protocol (RIP) in routers.

TOPOLOGY:



PROCEDURE:

- create a network with 3 routers, 2 switches and 4 end devices as shown in the topology above.
- configure the terminals of all the end devices with their respective IPs and set their gateways.
- encapsulate all the routers with Point to Point - PPP protocol along with a clock rate of 64000 while configuring them in the CLI
 - g the IPs of them in the CLI
 - > ip address x.x.x.x 255.0.0.0
 - > encapsulation PPP
 - > clock rate 64000
 - > no shutdown
 - > exit.

In the config terminals of the routers establish the RIP protocol by
(config) > router rip

> network [CONNECTED NETWORK'S IPs]

and exit

> now ping end device of 40's network from 10s.

OBSERVATION

> ping 40.0.0.1

Reply from 40.0.0.1 with 32 bytes of data

Reply from 40.0.0.1 : bytes = 32 time = 2ms TTL=125

Reply from 40.0.0.1 : bytes = 32 time = 2ms TTL=125

Reply from 40.0.0.1 : bytes = 32 time = 2ms TTL=125

Reply from 40.0.0.1 : bytes = 32 time = 8ms TTL=125

~~Ping statistics for 40.0.0.1 :~~

~~_packets: Sent = 4, Received = 4; Lost = 0~~

~~(0% loss)~~

Approximate round trip in milliseconds

Minimum = 2ms, Maximum = 40ms,

Average = 10ms

Since RIP protocol has been established,

IP route does not have to be set for each router.

Before RIP was set:

ping 10.0.0.1 → 40.0.0.1 : Destination host unreachable

Before RIP.

ping $10.x \rightarrow 20.x$

↳ Request Timed Out.

AIM:

Only on correctly configuring gateways and protocol, does the reply propagate & received properly

Topology

RESULT :-

(Routing Information Protocol) RIP is established in the network correctly.

NOTE: Even on proper connection and configuration, the first packet of the first internetwork ping is timed out as the switches have not learnt the network yet.

PROCEDURE

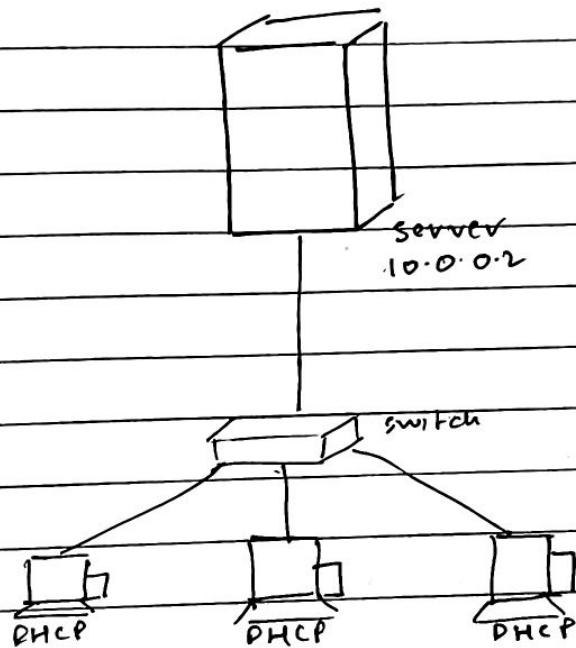
RIP - Routing Information Protocol

- is a dynamic routing protocol that uses hop count as a metric to find the best path between source and destination network.

~~8/8/12~~

AIM: Demonstration of web server and DNS using packet tracer.

Topology:



PROCEDURE :

Create a network as shown in the above topology. with one server, one switch and three end devices - PC₁, PC₂, PC₃

Configure the IP address of the server to 10.0.0.2 in the config window

Set starting IP as 10.0.0.1 in service window.

In the config windows of each PC_i, click on the DHCP radio button.

This dynamically allocates IP address to each of them

RESULT:

ping one device from another after waiting for IP to be allocated dynamically.

> ping 10.0.0.4

Pinging 10.0.0.4 with 32 bytes of data:

Reply from 10.0.0.4: bytes=32 time=4ms TTL=128

Reply from 10.0.0.4: bytes=32 time=5ms TTL=128

Reply from 10.0.0.4: bytes=32 time=0ms TTL=128

Reply from 10.0.0.4: bytes=32 time=4ms TTL=128

Ping statistics for 10.0.0.4:

Packets: sent = 4, received = 4, loss = 0
(0% loss)

Approximate round trip in milliseconds:

Minimum = 0ms

Maximum = 5ms

Average = 3ms

Neelam
15/12/2022

How to
server ?

Topology

PROCED

ce

above

co

enable

combi

In

Neelam
15/12/2022
it
address
Add

In t

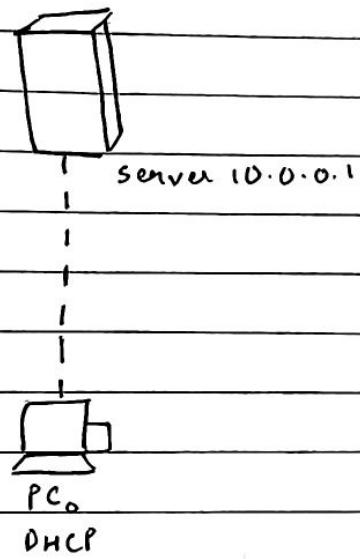
PC i,

Rf 8u

net

How to configure web server and DNS server ?

Topology



128
128
128
128

PROCEDURE :

create a network from the topology above

configure the IP of the server and enable DNS service, HTTP and DHCP
and enable DHCP on the end device

In services window click on DNS, create new.
it and type a name, ex 'bmico.com'
in the name and give it an IP address.

Add this to the list.

In the web browser application of the PC, type www.bmico.com

RESULT .

web server and DNS configured successfully

» Write a program for error detection using CRC (16-bit)

class CRC

{

void div(int a[], int k)

{

int gp[] = {1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1,
0, 0, 0, 0, 1};

int count = 0;

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

{

if (a[i] == gp[0])

{

for (int j = i; j < (i + k); j++)

{

a[j] = a[j] ^ gp[count++];

}

count = 0;

}

j

}

public static void main(String args [])

{

int a[] = new int[50];

int b[] = new int[50];

int len, k;

CRC ob = new CRC();

System.out.println("Enter length of data");

Scanner sc = new Scanner(System.in);

len = sc.nextInt();

int flag = 0;

breakdown

0,0,0,0,0,1,

```
sopen ("Enter data");
for (int i = 0; i < len; i++)
    a[i] = scannextInt();
for (int i = 0; i < 16; i++)
    a[len + i] = 0;
x = len - 16;
for (i = 0; i < len; i++)
    b[i] = a[i];
ob.div(a, k);
for (int i = 0; i < len; i++)
    a[i] = a[i] ^ b[i];
sopen ("Data to be transmitted ");
for (i = 0; i < len; i++)
    a[i] = a[i] ^ b[i];
System.out.println ("Data to be transmitted ");
for (int i = 0; i < len; i++)
    a[i] = sc.nextInt();
ob.div(a, k);
for (int i = 0; i < len; i++)
{
    if (a[i] == 0)
        flag = 1;
    break;
}
if (flag == 1)
    System.out.println ("Error");
else
    System.out.println ("No error");
```

3

}

Output:

→ Enter length of data from:

?

Enter the message

1011101

Data to be transmitted

10111011 0001 01101011 000

no error

Enter received data

1011011 000 101101 011 000

Error in data

31/2021

```
#include <stdio.h>
using namespace std;
int main()
{
    int no_of_queries, storage, output_pkt_size,
        input_pkt_size, bucket_size, size_left;
    storage = 0;
    printf("Enter the number of queries");
    scanf("%d", &no_of_queries);
    bucket_size = 10,
    output_pkt_size = 1;
    for (int i = 0; i < no_of_queries, i++)
    {
        printf("Enter the input size : \n");
        scanf("%d", &input_pkt_size);
        size_left = bucket_size - storage;
        if (input_pkt_size <= size_left)
            storage += input_pkt_size;
        else
            printf("Packet loss = %d\n",
                   input_pkt_size);
        printf("Buffer size = %d out of
               bucket size = %d\n",
               storage, bucket_size);
        storage -= output_pkt_size;
    }
    return 0;
}
```

Output

Buff-size = 20

Output-size = 5.

input size 15

Buffer-contents = 10

input size = 20

Buffer OVERFLOW.

N
S/1/2023

Dijkstra's algorithm for shortest path
for a given topology:

```
#include < stdio.h >
```

```
#include <conio.h>
```

```
define INFINITY 9999
```

```
#define MAX 100
```

```
void dijkstra(int G[MAX][MAX], int n,  
int startnode);
```

```
int main()
```

```
int G[MAX][MAX], j, i, n, u;  
int visited[MAX], count, middistance;  
printf("Enter number of vertices");  
scanf("%d", &n);  
printf("Enter the adjacency matrix");  
for (i = 0; i < n; i++)  
    for (j = 0; j < n; j++)  
        scanf("%d", &G[i][j]);
```

dijkstra(G, n, u) → 3 inside for loop

```
return 0;
```

```
}
```

```
void dijkstra (int G[MAX][MAX], int n, int startnode)
```

```
int count [MAX][MAX], distance [MAX], pred [MAX]  
int count, min distance, visited [10],  
nextnode, i, j
```

```

for (i=0; i<n; i++)
  for (j = 0; j < n; j++)
    if (G[i][j] == -∞)
      cost[i][j] == INFINITY
    else
      cost[i][j] = G[i][j]
  
```

```

for (i=0; i<n; i++)
  {
    dist[i] = cost[startnode][i];
    pred[i] = startnode;
    visited[i] = 0;
  }

```

```

distance[startnode = 0];
visited[startnode = 1];
count = 1;

```

```

while
  for (count < n-1)
    {

```

```

      mindistance = INFINITY;
      for (i=0; i<n; i++)
        {
          if (dist[i] < mindistance & !vis[i])
            {
              mindist = dist[i];
              nextnode = i;
            }
        }
    }

```

```

  visited[nextnode] = 1

```

```

  visited[nextnode] = 1

```

```

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

```

```

      if (!visited[i])
        {

```

```

          if (mindistance + cost[nextnode][i]
              < distance[i])
            {

```

```

              distance[i] = mindist +

```

```

              cost[nextnode][i]
            }

```

```

            pred[nextnode];
          }
        }
      }
    }
  }

```

3) count ++;

int shortest
only

ed (array x)

(10),

for

for ($i = 0; i < n; i++$)

 if ($i \neq \text{start node}$)

 printf ("\\n Distance of v. d = $v.d$,
 $i, \text{distance}[i]$

 printf ("\\n Path = $v.d[i]$);

$j = i$

 int count = 1

 do

$j = \text{pred}[j]$

 printf (" $\leftarrow v.d$ ", j)

 count ++;

 } while ($j \neq \text{start node}$);

 printf ("\\n Count = $v.d[i]$ ", count);

 }

 }

}

OUTPUT:-

Enter no. of vertices = 4

Enter adjacency matrix:

0 1 1,

1 0 1 0

1 1 0 1

1 0 1 0

Enter starting node ,

Distance of 0 = 1

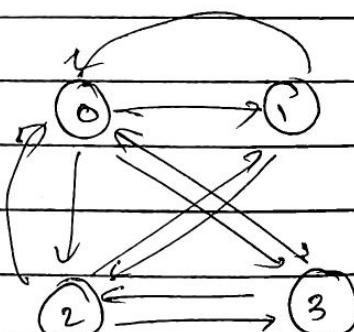
path = 0 \leftarrow 1

Distance of 2 = 1

path = 2 \leftarrow 1

Distance of 3 = 2

path = 3 \leftarrow 0 \leftarrow 1



write a program for distance vector algorithm to find suitable path for transmission.

```

#include <stdio.h>
#include <stdlib.h>
int Bellman-Ford() int G[20][20], int v,
int E, int edge[20][2]
{
    int d, u, v, k, dist[20], parent[20],
    , flag = 1;
    for (i=0; i<v, i++)
        {distance [i]} = 1000, parent [i] = -1;
    printf("Enter source ");
    scanf("%d", &s);
    for (k=0; k < v-1; k++)
    {
        for (u = 0; u < E; u++)
        {
            v = edge [k][0], w = edge [k][1];
            if (distance [u] + G[u][v] <
                distance [v])
                distance [v] = distance [u] +
                G[u][v];
            parent [v] = u;
        }
    }
    for (k = 0; k < E; k++)
    {
        u = edge [k][0], v = edge [k][1];
        if (distance [u] + G[u][v] < distance [v])
        {
            flag = 0
        }
    }
}

```

if (flag)

for (i = 0; i < v; i++)

 parent ("vertex " + i + " → cost = " + cost[i])

 parent = " " + i + " " + i + 1 + ", distance[" + i + "])

 parent(i) + 1);

 return flag;

}

OUTPUT :-

Bellman Ford

Enter no vertices : 5

Enter graph in matrix

0 2 2 99 99 ,

2 0 99 3 99

2 99 0 6 4

99 3 6 6 9

99 99 4 5 0

enter source = 1

Vertex 1 → cost = 0 parent = 0

Vertex 2 → cost = 2 parent = 1

Vertex 3 → cost = 2 parent = 1

Vertex 4 → cost = 5 parent = 2

Vertex 5 → cost = 6 parent = 3

No negative weight edge.

using TCP/IP sockets , write a client server program to make client sending the filename and the server to send back the contents of the required file if present.

Client :-

```
from socket import *
serverName = 127.0.0.1
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName, serverPort))
sentence = input("Enter file name:")
clientSocket.send(sentence.encode())
filecontents = clientSocket.recv(1024).decode(),
print("From server : " + sentence)
print(filecontents)
clientSocket.close()
```

Server :

```
from socket import *
serverName = 127.0.0.1
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind((serverName, serverPort))
serverSocket.listen(1)
while 1:
    print("Server is ready to receive")
```

```
connectionSocket, addr = serverSocket.accept()
sentence = connectionSocket.recv(1024).decode()
file = open(sentence + '.txt', 'w')
l = file.read(1024)
connectionSocket.send(l.encode())
print("Content of " + sentence + ".txt is")
file.close()
connectionSocket.close()
```

OUTPUT:

```
Enter filename :- Server TCP.py
from socket import *
serverName = '127.0.0.1'
serverPort = 12000
connectionSocket.send(l.encode())
file.close()
connectionSocket.close()
```

EXPERIMENT - 13

PAGE NO.

DATE

Using UDP - client server program :-

client :

```

from socket import *
serverName = "127.0.0.0"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
sentence = input("Enter file name: ")
clientSocket.sendto(sentence.encode(), (serverName, serverPort))
fileContent, serverAddress = clientSocket.recvfrom(2048)
print(fileContent.decode())
clientSocket.close()
    
```

server

```

from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("128.0.0.1", serverPort))
print("Server is ready to receive")
while True:
    sentence, clientAddress = serverSocket.recvfrom(2048)
    sentence = sentence.decode("utf-8")
    file = open(sentence, "r")
    l = file.read(2048)
    serverSocket.sendto(l, clientAddress)
    print(sentence)
    file.close()
    
```

OUR OUT:

Enter filename : server UDP.py

Reply from server :

from socket import *

serverPort = 12000

serverSocket.bind(('127.0.0.1', serverPort))

serverSocket.sendto((sys.argv[1], "utf-8"))

clientAddress

filename).