**BMS College of Engineering**

**Department of CSE**

**V SEMESTER - Aug-Dec 2017**

**Course Title:** COMPUTER NETWORKS  **Course Code: 16CS5DCCON**

**LIST OF LAB PROGRAMS**

**CYCLE 1**

1. **How to configure DHCP within a LAN in a packet Tracer**

**SOLUTION:**

Here,we are going to see how to configure DHCP within a LAN in a packet tracer, DHCP-Dynamic Host Configuration Protocol DHCP Proces,(DORA) =>Discover( Client discovers the DHCP Server for Ip request ) =>Offer ( DHCP servers offers ip to a client and it will wait for the request from cleint =>Request ( Client reply for requesting offered ip address from dhcp server,sometimes there will be more dhcp servers.So,client receives offer from multiple servers.client will reply for only one offer.So, other offers will be canceled and replaced in a pool.) =>Acknowledgement (Final step from server sending all infromation to client like DNS,Default Gateway,tftp,domain name,... ) DHCP Server uses Port 67,Client uses port 68 It uses UDP(User Datagram Protocol). **Step 1:**Create a LAN like this,



**Step 1:**Create a LAN like this,

Router>enable Router

#config t Router(config)

#interface fastethernet0/0 Router(config-if)

#ip address 10.0.0.1 255.0.0.0 Router(config-if)

#no shutdown Router(config-if)

#exit Router(config)

**Step 3:**click on server-> config, then assign gateway in our example 10.0.0.1

**Step 4:**Then Click on Fastethernet and assign ip address and subnet mask.I am going to use 10.0.0.2 and subnet mask 255.0.0.0 for our server.

**Step 5:** Click on DHCP,there you can see default pool**,**

**Step 6:**Just give default gate way,here we are using 10.0.0.1.

**Step 7:**DNS server,Just give our server ip address,10.0.0.2.

**Step 8:**Then just edit start ip address.I am going to give 10.0.0.10 and subnet mask 255.0.0.0 **Step 9:**In Maximum Number of Users,Here we are using Class A Network so we can use 1,67,77,216 ip address.just give how many ip address you want in this pool.I am going to give 500

**Step 10:**Assign TFTP server ip address,just give our server ip address,10.0.0.2.

**Step 11:** And click on save.That's it...



**Step 12:**Now, Click on any of the PC-> then click on Desktop->Ip configuration,and Choose 'DHCP' wait for some time,if your dhcp request failed then try few more times.This is how you should get.



1. How to get the IP from DHCP that is present in some other Network using IP helper Address.

**SOLUTION:**

Here we are going to see, how to configure DHCP for multiple netowrks .Can we get ip address from DHCP that is prensent in other network? yes we can.Lets see how to do with help of 'ip helper-address'.

**Step 1:**Create a topology like this,

**Step 2:**Configure the router

interface fastethernet0/0 and fastethernet 0/1 with ip address .

Router>enable Router

#config terminal Router(config)

#interface fastethernet0/0 Router(config-if)

#ip address 10.0.0.1 255.0.0.0 Router(config-if)

#no shutdown Router(config-if)

#exit Router(config)

#interface fastethernet0/1 Router(config-if)

#ip address 20.0.0.1 255.0.0.0 Router(config-if)

#no shutdown Router(config-if)

#exit



**Step 3:**Click on server->config->then just give the gateway ip address .Gateway for this network is 10.0.0.1

**Step 4:**Then click on fastethernet assign ip address.I am going to give 10.0.0.2 and subnetmask 255.0.0.0.Once we have configured the ip address for the server,DHCP server automatically assign 10 network for default pool.We don't have to create pool for 10 Network again.Just we need to give ip for DNS,Gateway and TFTP then we may configure starting ip address or leave it and Save. **Step 5:**Now,Click on Pc in a LAN with Server and Check whether DHCP working fine in this network.Click on any PC->Desktop->Ip configuration->Choose DHCP, then you will get ip from dhcp server for this PC.



**Step 6:**Now,we see how to get ip address for PC that is in a network without Server.For that, first we have to add network pool in a dhcp server. So,Click on Server->Config->DHCP.

**Step 7:**Just edit Pool Name with any other name.I am going to give 20Network. Default Gateway->20.0.0.1, DNS Server->10.0.0.2

Start Ip Address->20.0.0.10 Subnet Mask->255.0.0.0 Maximun Number Of Users->100 TFTP Server10.0.0.2 Then,Click on Add and Save.

**Step 8**: Now go to router and give ip helper address under fastethernet0/1,that is server address here our server address is 10.0.0.2.Now we can get ip for this network also In Router,(Global configuration mode)

Router(config)#interface fastethernet0/1

Router(config-if)#ip helper-address 10.0.0.2

Router(config-if)#exit

**Step 9:**Now,check whether PC from network without server getting ip from the DHCP server in another Network.Click on any PC->Desktop->Ip configuration->Choose DHCP. Now we have got ip address from dhcp server.

1. How to configure RIP Routing Protocol in Routers

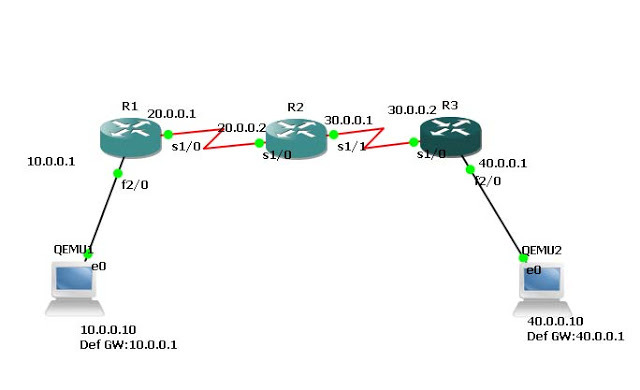
**SOLUTION:**

RIP-Routing Information Protocol

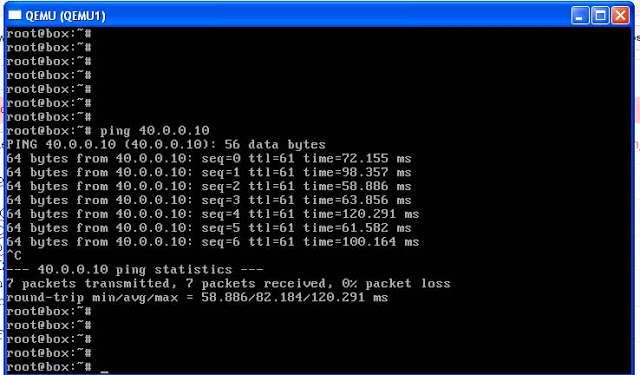
Routing Information Protocol is distance vector routing protocol.It knows only neighbors,

It doesn't know entire topology,Routing by rumors  
It doesn't support classless network(CIDR),But Rip Version 2 supports CIDR and VLSM  
It will update routing informations every 30 seconds(Periodic Updates)  
It sends entire routing table to its neighbour  
  
RIP- Routing Information Protocol  
Administrative Distance-120  
Maximum Hop count-15,16 is unreachable network  
Metric - Hop count

Here we are going to see how to configure RIP in a simple topology, check connectivity between hosts

**Step 1**:Create a topology like i have added below  
[](http://2.bp.blogspot.com/-NCkyHW-tn5s/UQ9X0-0mSWI/AAAAAAAAAF0/wb1Hx4w65Sc/s1600/1.jpg)

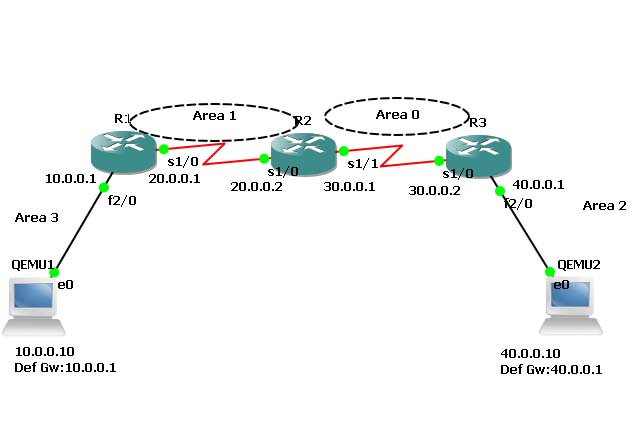
**Step 2**:Configure ip address for all interfaces and assign ip address,default gateway to hosts  
  
**In Router R1,**  
  
**R1(config)#interface fastethernet 2/0**  
**R1(config-if)#ip address 10.0.0.1 255.0.0.0**  
**R1(config-if)#no shutdown**  
**R1(config-if)#exit**  
 **R1(config)#interface serial 1/0**  
**R1(config-if)#ip address 20.0.0.1 255.0.0.0**  
**R1(config-if)#encapsulation ppp**  
**R1(config-if)#clock rate 64000**  
**R1(config-if)#no shutdown**  
**R1(config-if)#exit**  
 **In Router R2,**  
 **R2(config)#interface serial 1/0**  
**R2(config-if)#ip address 20.0.0.2 255.0.0.0**  
**R2(config-if)#encapsulation ppp**  
**R2(config-if)#no shutdown**  
**R2(config-if)#exit**  
 **R2(config)#interface serial 1/1**  
**R2(config-if)#ip address 30.0.0.1 255.0.0.0**  
**R2(config-if)#encapsulation ppp**  
**R2(config-if)#clock rate 64000**  
**R2(config-if)#no shutdown**  
**R2(config-if)#exit**  
 **In Router R3,**  
 **R3(config)#**  
**R3(config)#interface serial 1/0**  
**R3(config-if)#ip address 30.0.0.2 255.0.0.0**  
**R3(config-if)#encapsulation ppp**  
**R3(config-if)#no shutdown**  
**R3(config-if)#exit**  
 **R3(config)#**  
**R3(config)#interface fastethernet 2/0**  
**R3(config-if)#ip address 40.0.0.1 255.0.0.0**  
**R3(config-if)#no shutdown**  
**R3(config-if)#exit**  
  
**Step 3**:Configure RIP to all routers by using command,  
**In Router R1,**  
 **R1(config)#router rip**  
**R1(config-router)#network 10.0.0.0**  
**R1(config-router)#network 20.0.0.0**  
**R1(config-router)#exit**  
  
**In Router R2,**  
 **R2(config)#router rip**  
**R2(config-router)#network 20.0.0.0**  
**R2(config-router)#network 30.0.0.0**  
**R2(config-router)#exit**  
 **In Router R3,**  
 **R3(config)#router rip**  
**R3(config-router)#network 30.0.0.0**  
**R3(config-router)#network 40.0.0.0**  
**R3(config-router)#exit**  
 **Step 4**: Now check routing table of  route R1.Router will have all network information in its routing table, router learned this route by using rip.  
  
**R1#show ip route**  
**Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP**  
**D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area**  
**N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2**  
**E1 - OSPF external type 1, E2 - OSPF external type 2**  
**i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area**  
**\* - candidate default, U - per-user static route, o - ODR**  
**P - periodic downloaded static route**  
 **Gateway of last resort is not set**  
 **C    10.0.0.0/8 is directly connected, FastEthernet2/0**  
**C    20.0.0.0/8 is directly connected, Serial1/0**  
**R    40.0.0.0/8 [120/2] via 20.0.0.2, 00:00:02, Serial1/0**  
**R    30.0.0.0/8 [120/1] via 20.0.0.2, 00:00:02, Serial1/0**  
  
  
**Step 5**:Now ping from host 10.0.0.10 to 40.0.0.10

[](http://4.bp.blogspot.com/-sNR7Ur52nZo/UQ9dWbV0SUI/AAAAAAAAAGE/WvOT42Zvk14/s1600/2.jpg)

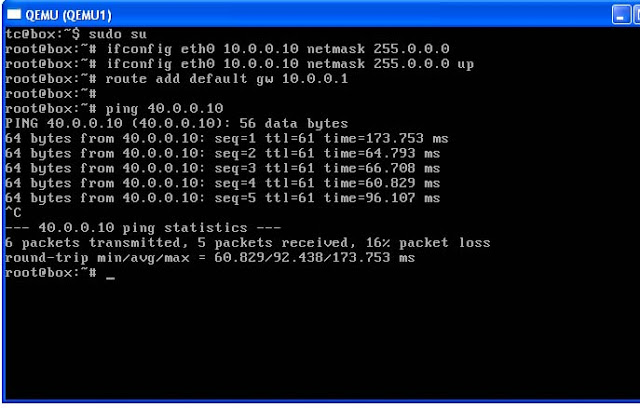
RIP Version 2  
  
RIP- Routing Information Protocol  
Administrative Distance-120  
Maximum Hop count-15,16 is unreachable network  
Metric - Hop count  
  
Rip version 2 supports CIDR(Classless Ineternet Domain Routing)  
  
To configure with rip version 2,just give this command,  
 **R1(config)#router rip**  
**R1(config-router)#version 2**  
**R1(config-router)#network 10.0.0.0**  
**R1(config-router)#network 20.0.0.0**  
**R1(config-router)#exit**

# [How to configure OSPF Routing Protocol and Connect Areas](http://chennaicisco.blogspot.in/2013/02/how-to-configure-ospf-routing-protocol.html)

**OSPF-Open Short Path First**,It is a Open Standard Routing Protocol  
  
Link State Routing Protocol  
Algorithm -Dijkistra,To find shortest path  
Administrativ Distance-110  
Metric- 10^8/Bandwidth  
Incremental Update  
Load Balancing Maximum 6 (Default 4)  
It knows Network topology  
Router ID-Initializes OSPF Process  
Fast convergence  
  
Timers  
Hello-10  
Dead-40  
Wait-40  
Retransmit-5  
  
**Step 1**:Create topology like below i have given

[](http://1.bp.blogspot.com/-5eRJ-YVs_Z4/URcfDZmLSeI/AAAAAAAAAGk/v1HEyPuVP0A/s1600/1.jpg)

**Step 2**:Cofigure ip addres to all interfaces  
  
 **In Router R1,**  
 **R1(config)#interface fastethernet 2/0**  
**R1(config-if)#ip address 10.0.0.1 255.0.0.0**  
**R1(config-if)#no shutdown**  
**R1(config-if)#exit**  
 **R1(config)#interface serial 1/0**  
**R1(config-if)#ip address 20.0.0.1 255.0.0.0**  
**R1(config-if)#encapsulation ppp**  
**R1(config-if)#clock rate 64000**  
**R1(config-if)#no shutdown**  
**R1(config-if)#exit**  
 **In Router R2,**  
 **R2(config)#interface serial 1/0**  
**R2(config-if)#ip address 20.0.0.2 255.0.0.0**  
**R2(config-if)#encapsulation ppp**  
**R2(config-if)#no shutdown**  
**R2(config-if)#exit**  
 **R2(config)#interface serial 1/1**  
**R2(config-if)#ip address 30.0.0.1 255.0.0.0**  
**R2(config-if)#encapsulation ppp**  
**R2(config-if)#clock rate 64000**  
**R2(config-if)#no shutdown**  
**R2(config-if)#exit**  
 **In Router R3,**  
 **R3(config)#**  
**R3(config)#interface serial 1/0**  
**R3(config-if)#ip address 30.0.0.2 255.0.0.0**  
**R3(config-if)#encapsulation ppp**  
**R3(config-if)#no shutdown**  
**R3(config-if)#exit**  
 **R3(config)#**  
**R3(config)#interface fastethernet 2/0**  
**R3(config-if)#ip address 40.0.0.1 255.0.0.0**  
**R3(config-if)#no shutdown**  
**R3(config-if)#exit**  
  
**Step 3**:Now,Enable ip routing by configuring ospf routing protocol in all routers,  
  
**In Router R1,**  
**R1(config)#router ospf 1**  
**R1(config-router)#router-id 1.1.1.1**  
**R1(config-router)#network 10.0.0.0 0.255.255.255 area 3**  
**R1(config-router)#network 20.0.0.0 0.255.255.255 area 1**  
**R1(config-router)#exit**  
 **In Router R2,**  
**R2(config)#router ospf 1**  
**R2(config-router)#router-id 2.2.2.2**  
**R2(config-router)#network 20.0.0.0 0.255.255.255 area 1**  
**R2(config-router)#network 30.0.0.0 0.255.255.255 area 0**  
**R2(config-router)#exit**  
 **In Router R3,**  
**R3(config)#router ospf 1**  
**R3(config-router)#router-id 3.3.3.3**  
**R3(config-router)#network 30.0.0.0 0.255.255.255 area 0**  
**R3(config-router)#network 40.0.0.0 0.255.255.255 area 2**  
**R3(config-router)#exit**  
  
You have to configure router id when we configure ospf.It is used to identify the router  
  
**Step 4**:Now check routing table of R1,  
  
**Router#show ip route**  
**Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP**  
**D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area**  
**N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2**  
**E1 - OSPF external type 1, E2 - OSPF external type 2**  
**i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area**  
**\* - candidate default, U - per-user static route, o - ODR**  
**P - periodic downloaded static route**  
 **Gateway of last resort is not set**  
 **C    10.0.0.0/8 is directly connected, FastEthernet2/0**  
**C     20.0.0.0/8 is directly connected, Serial1/0**  
**O IA 40.0.0.0/8 [110/129] via 20.0.0.2, 00:04:23, Serial1/0**  
**O IA 30.0.0.0/8 [110/128] via 20.0.0.2, 00:07:29, Serial1/0**  
  
Here,R2 knows Area 0.Network 20.0.0.0 connected to R2 from R1,So R1 learns networks through this network.  
  
R3(config)#router ospf 1, Here, 1 is Process ID, it can be 1-65535.It initializes ospf process.  
            
There must be one interface up to keep ospf process up.So its better to configure loopback address to routers.It is a virtual interface never goes down once we configured.  
  
**R1(config-if)#interface loopback 0**  
**R1(config-if)#ip add 172.16.1.252 255.255.0.0**  
**R1(config-if)#no shutdown**  
  
**R2(config-if)#interface loopback 0**  
**R2(config-if)#ip add 172.16.1.253 255.255.0.0**  
**R2(config-if)#no shutdown**  
  
**R3(config-if)#interface loopback 0**  
**R3(config-if)#ip add 172.16.1.254 255.255.0.0**  
**R3(config-if)#no shutdown**  
  
**Step 5**:Now ,Check Routing table of R3,  
  
**R3#show ip route**  
**Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP**  
**D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area**  
**N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2**  
**E1 - OSPF external type 1, E2 - OSPF external type 2**  
**i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area**  
**\* - candidate default, U - per-user static route, o - ODR**  
**P - periodic downloaded static route**  
 **Gateway of last resort is not set**  
 **O IA 20.0.0.0/8 [110/128] via 30.0.0.1, 00:18:58, Serial1/0**  
**C    40.0.0.0/8 is directly connected, FastEthernet2/0**  
**C    30.0.0.0/8 is directly connected, Serial1/0**  
  
Here,R3 doesn't know about the area 3 so we have to create virtual link between R1 and R2  
  
**Step 6**:Create virtual link between R1,R2,by this we create a virtual link to connect area 3 to area 0.  
  
**In Router R1,**  
**R1(config)#router ospf 1**  
**R1(config-router)#area 1 virtual-link 2.2.2.2**  
**R1(config-router)#**  
**\*Feb 10 10:29:23.767: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on OSPF\_VL0 from LOADING to FULL, Loadi**  
**ng Done**  
 **In Rotuer R2,**  
**\*Feb 10 10:28:59.543: %OSPF-4-ERRRCV: Received invalid packet: mismatch area ID, from backbone area must be virtual-link but not found from 20.0.0.1, Serial1/0a**  
**\*Feb 10 10:29:09.535: %OSPF-4-ERRRCV: Received invalid packet: mismatch area ID, from backbone area must be virtual-link but not found from 20.0.0.1, Serial1/0.1.1**  
 **R2(config-router)#**  
**R2(config-router)#area 1 virtual-link 1.1.1.1**  
**R2(config-router)#exit**  
**R2(config)#**  
 **\*Feb 10 10:29:19.667: %OSPF-5-ADJCHG: Process 1, Nbr 1.1.1.1 on OSPF\_VL0 from LOADING to FULL, Loadi**  
**ng Done**  
  
**Step 7**:R2 and R3 get updates about Area 3 .Now,Check routing table of R3,  
  
**R3#show ip route**  
**Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP**  
**D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area**  
**N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2**  
**E1 - OSPF external type 1, E2 - OSPF external type 2**  
**i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area**  
**\* - candidate default, U - per-user static route, o - ODR**  
**P - periodic downloaded static route**  
 **Gateway of last resort is not set**  
 **O IA 20.0.0.0/8 [110/128] via 30.0.0.1, 00:01:56, Serial1/0**  
**C    40.0.0.0/8 is directly connected, FastEthernet2/0**  
**O IA 10.0.0.0/8 [110/129] via 30.0.0.1, 00:01:56, Serial1/0**  
**C     30.0.0.0/8 is directly connected, Serial1/0**  
  
**Step 8**:Check connectivity between host 10.0.0.10 to 40.0.0.10

[](http://1.bp.blogspot.com/-380Fi5zXE6I/URcr0l82R_I/AAAAAAAAAG0/HUkaaZ-RyMU/s1600/2.jpg)

# Trouble shooting R3#show ip protocols R3#show ip route ospf R3#show ip ospf neighbors detail R3#show ip ospf database R3#show ip ospf interface

**CYCLE 2**

1. Create a simple topology of three nodes (Node1, Node2, Node3) separated by a point-to-point link. Setup a UdpClient on Node1 and a UdpServer on Node2. Let the data rate be set 5 Mbps and the delay be 2 ms. Now Setup another UdpClient on Node3 and a server instance on Node2. Let the data rate be set 7 Mbps and the delay be 1 ms. Set the parameters for the clients. Run the simulation and observe the results.

**SOLUTION:**

#include "ns3/core-module.h"

#include "ns3/network-module.h"

#include "ns3/internet-module.h"

#include "ns3/point-to-point-module.h"

#include "ns3/applications-module.h"

using namespace ns3;

NS\_LOG\_COMPONENT\_DEFINE ("FirstScriptExample");

int main (int argc, char \*argv[])

{

Time::SetResolution (Time::NS);

LogComponentEnable ("UdpEchoClientApplication", LOG\_LEVEL\_INFO);

LogComponentEnable ("UdpEchoServerApplication", LOG\_LEVEL\_INFO);

NodeContainer nodes;

nodes.Create (2);

NodeContainer nodes1;

nodes1.Create (2);

PointToPointHelper pointToPoint;

pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));

pointToPoint.SetChannelAttribute ("Delay", StringValue ("2ms"));

PointToPointHelper pointToPoint1;

pointToPoint1.SetDeviceAttribute ("DataRate", StringValue ("7Mbps"));

pointToPoint1.SetChannelAttribute ("Delay", StringValue ("1ms"));

NetDeviceContainer devices;

devices = pointToPoint.Install (nodes);

NetDeviceContainer devices1;

devices1 = pointToPoint1.Install (nodes1);

InternetStackHelper stack;

stack.Install (nodes);

stack.Install (nodes1);

Ipv4AddressHelper address,address1;

address.SetBase ("10.1.1.0", "255.255.255.0");

address1.SetBase ("10.1.2.0", "255.255.255.0");

Ipv4InterfaceContainer interfaces = address.Assign (devices);

Ipv4InterfaceContainer interfaces1 = address1.Assign (devices1);

UdpEchoServerHelper echoServer (9);

ApplicationContainer serverApps = echoServer.Install (nodes.Get (1));

serverApps.Start (Seconds (1.0));

serverApps.Stop (Seconds (10.0));

UdpEchoClientHelper echoClient (interfaces.GetAddress (1), 9);

echoClient.SetAttribute ("MaxPackets", UintegerValue (1));

echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.0)));

echoClient.SetAttribute ("PacketSize", UintegerValue (1024));

ApplicationContainer clientApps = echoClient.Install (nodes.Get (0));

clientApps.Start (Seconds (2.0));

clientApps.Stop (Seconds (10.0));

UdpEchoServerHelper echoServer1 (10);

ApplicationContainer serverApps1 = echoServer1.Install (nodes1.Get (0));

serverApps1.Start (Seconds (1.0));

serverApps1.Stop (Seconds (10.0));

UdpEchoClientHelper echoClient1 (interfaces1.GetAddress (1), 10);

echoClient1.SetAttribute ("MaxPackets", UintegerValue (2));

echoClient1.SetAttribute ("Interval", TimeValue (Seconds (2.0)));

echoClient1.SetAttribute ("PacketSize", UintegerValue (512));

ApplicationContainer clientApps1 = echoClient1.Install (nodes1.Get (1));

clientApps1.Start (Seconds (2.0));

clientApps1.Stop (Seconds (10.0));

Simulator::Run ();

Simulator::Destroy ();

return 0;

}

1. **Create a topology as given below**

**10.1.1.0 10.1.3.0**

**n0 -------------- n1 n2 n3 n4 -------------- n5**

**point-to-point | | | | point-to-point**

**=============**

**LAN 10.1.2.0**

**Choose a node from point-to-point network as a client and one from the LAN as server. Run the simulation and observe the results.**

**SOLUTION:**

#include "ns3/core-module.h"

#include "ns3/network-module.h"

#include "ns3/csma-module.h"

#include "ns3/internet-module.h"

#include "ns3/point-to-point-module.h"

#include "ns3/applications-module.h"

#include "ns3/ipv4-global-routing-helper.h"

// Default Network Topology

//

// 10.1.1.0 10.1.3.0

// n0 -------------- n1 n2 n3 n4 -------------- n5

// point-to-point | | | | point-to-point

// ================

// LAN 10.1.2.0

using namespace ns3;

NS\_LOG\_COMPONENT\_DEFINE ("SecondScriptExample");

int

main (int argc, char \*argv[])

{

bool verbose = true;

uint32\_t nCsma = 2;

CommandLine cmd;

cmd.AddValue ("nCsma", "Number of \"extra\" CSMA nodes/devices", nCsma);

cmd.AddValue ("verbose", "Tell echo applications to log if true", verbose);

cmd.Parse (argc,argv);

if (verbose)

{

LogComponentEnable ("UdpEchoClientApplication", LOG\_LEVEL\_INFO);

LogComponentEnable ("UdpEchoServerApplication", LOG\_LEVEL\_INFO);

}

nCsma = nCsma == 0 ? 1 : nCsma;

NodeContainer p2pNodes;

p2pNodes.Create (2);

NodeContainer p2pNodes1;

p2pNodes1.Create(2);

NodeContainer csmaNodes;

csmaNodes.Add (p2pNodes.Get (1));

csmaNodes.Create (nCsma);

csmaNodes.Add(p2pNodes1.Get(0));

PointToPointHelper pointToPoint;

pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));

pointToPoint.SetChannelAttribute ("Delay", StringValue ("2ms"));

CsmaHelper csma;

csma.SetChannelAttribute ("DataRate", StringValue ("100Mbps"));

csma.SetChannelAttribute ("Delay", TimeValue (NanoSeconds (6560)));

PointToPointHelper pointToPoint1;

pointToPoint1.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));

pointToPoint1.SetChannelAttribute ("Delay", StringValue ("2ms"));

NetDeviceContainer p2pDevices;

p2pDevices = pointToPoint.Install (p2pNodes);

NetDeviceContainer csmaDevices;

csmaDevices = csma.Install (csmaNodes);

NetDeviceContainer p2pDevices1;

p2pDevices1 = pointToPoint1.Install (p2pNodes1);

InternetStackHelper stack;

stack.Install (p2pNodes.Get (0));

stack.Install (csmaNodes);

stack.Install (p2pNodes1.Get (1));

Ipv4AddressHelper address;

address.SetBase ("10.1.1.0", "255.255.255.0");

Ipv4InterfaceContainer p2pInterfaces;

p2pInterfaces = address.Assign (p2pDevices);

address.SetBase ("10.1.2.0", "255.255.255.0");

Ipv4InterfaceContainer csmaInterfaces;

csmaInterfaces = address.Assign (csmaDevices);

address.SetBase ("10.1.3.0", "255.255.255.0");

Ipv4InterfaceContainer p2pInterfaces1;

p2pInterfaces1 = address.Assign (p2pDevices1);

UdpEchoServerHelper echoServer (9);

ApplicationContainer serverApps = echoServer.Install (csmaNodes.Get (nCsma));

serverApps.Start (Seconds (1.0));

serverApps.Stop (Seconds (10.0));

UdpEchoServerHelper echoServer1 (10);

ApplicationContainer serverApps1 = echoServer1.Install (csmaNodes.Get (nCsma));

serverApps1.Start (Seconds (1.0));

serverApps1.Stop (Seconds (10.0));

UdpEchoClientHelper echoClient (csmaInterfaces.GetAddress (nCsma), 9);

echoClient.SetAttribute ("MaxPackets", UintegerValue (1));

echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.0)));

echoClient.SetAttribute ("PacketSize", UintegerValue (1024));

ApplicationContainer clientApps = echoClient.Install (p2pNodes.Get (0));

clientApps.Start (Seconds (2.0));

clientApps.Stop (Seconds (10.0));

UdpEchoClientHelper echoClient1 (csmaInterfaces.GetAddress (nCsma), 10);

echoClient.SetAttribute ("MaxPackets", UintegerValue (2));

echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.0)));

echoClient.SetAttribute ("PacketSize", UintegerValue (512));

ApplicationContainer clientApps1 = echoClient1.Install (p2pNodes1.Get (1));

clientApps1.Start (Seconds (2.0));

clientApps1.Stop (Seconds (10.0));

Ipv4GlobalRoutingHelper::PopulateRoutingTables ();

pointToPoint.EnablePcapAll ("second");

csma.EnablePcap ("second", csmaDevices.Get (1), true);

pointToPoint1.EnablePcapAll ("second");

//pointToPoint1.EnablePcap ("second\_new", p2pNodes1.Get (0), true);

Simulator::Run ();

Simulator::Destroy ();

return 0;

}

1. **Create a simple topology of two nodes separated by a point to point link. Demonstrate the use of FlowMonitor.**

**SOLUTION:**

#include <fstream>

#include "ns3/core-module.h"

#include "ns3/point-to-point-module.h"

#include "ns3/applications-module.h"

#include "ns3/internet-module.h"

#include "ns3/flow-monitor-module.h"

using namespace ns3;

NS\_LOG\_COMPONENT\_DEFINE ("Lab1");

int main (int argc, char \*argv[])

{

double lat = 2.0;

uint64\_t rate = 5000000; // Data rate in bps

double interval = 0.05;

CommandLine cmd;

cmd.AddValue ("latency", "P2P link Latency in miliseconds", lat);

cmd.AddValue ("rate", "P2P data rate in bps", rate);

cmd.AddValue ("interval", "UDP client packet interval", interval);

cmd.Parse (argc, argv);

// Explicitly create the nodes required by the topology (shown above).

NS\_LOG\_INFO ("Create nodes.");

NodeContainer n;

n.Create (2);

NS\_LOG\_INFO ("Create channels.");

// Explicitly create the channels required by the topology (shown above).

PointToPointHelper p2p;

p2p.SetChannelAttribute ("Delay", TimeValue (MilliSeconds (lat)));

p2p.SetDeviceAttribute ("DataRate", DataRateValue (DataRate (rate)));

p2p.SetDeviceAttribute ("Mtu", UintegerValue (1400));

NetDeviceContainer dev = p2p.Install (n.Get(0), n.Get(1));

//NetDeviceContainer dev2 = p2p.Install (n.Get(1), n.Get(2));

// We've got the "hardware" in place. Now we need to add IP addresses.

// Install Internet Stack

InternetStackHelper internet;

internet.Install (n);

Ipv4AddressHelper ipv4;

NS\_LOG\_INFO ("Assign IP Addresses.");

ipv4.SetBase ("10.1.1.0", "255.255.255.0");

Ipv4InterfaceContainer i = ipv4.Assign (dev);

/\*ipv4.SetBase ("10.1.2.0", "255.255.255.0");

Ipv4InterfaceContainer i2 = ipv4.Assign (dev2);\*/

Ipv4GlobalRoutingHelper::PopulateRoutingTables ();

NS\_LOG\_INFO ("Create Applications.");

// Create one udpServer application on node one.

uint16\_t port1 = 8000; // Need different port numbers to ensure there is no conflict

// uint16\_t port2 = 8001;

UdpServerHelper server1 (port1);

//UdpServerHelper server2 (port2);

ApplicationContainer apps;

apps = server1.Install (n.Get (1));

//apps = server2.Install (n.Get (2));

apps.Start (Seconds (1.0));

apps.Stop (Seconds (10.0));

// Create one UdpClient application to send UDP datagrams from node zero to

// node one.

uint32\_t MaxPacketSize = 1024;

Time interPacketInterval = Seconds (interval);

uint32\_t maxPacketCount = 320;

UdpClientHelper client1 (i.GetAddress (1), port1);

//UdpClientHelper client2 (i2.GetAddress (1), port2);

client1.SetAttribute ("MaxPackets", UintegerValue (maxPacketCount));

client1.SetAttribute ("Interval", TimeValue (interPacketInterval));

client1.SetAttribute ("PacketSize", UintegerValue (MaxPacketSize));

/\*client2.SetAttribute ("MaxPackets", UintegerValue (maxPacketCount));

client2.SetAttribute ("Interval", TimeValue (interPacketInterval));

client2.SetAttribute ("PacketSize", UintegerValue (MaxPacketSize));\*/

apps = client1.Install (n.Get (0));

//apps = client2.Install (n.Get (0));

apps.Start (Seconds (2.0));

apps.Stop (Seconds (10.0));

p2p.EnablePcap("lab-1", dev, false);

// Calculate Throughput using Flowmonitor

FlowMonitorHelper flowmon;

Ptr<FlowMonitor> monitor = flowmon.InstallAll();

// Now, do the actual simulation.

NS\_LOG\_INFO ("Run Simulation.");

Simulator::Stop (Seconds(11.0));

Simulator::Run ();

monitor->CheckForLostPackets ();

Ptr<Ipv4FlowClassifier> classifier = DynamicCast<Ipv4FlowClassifier> (flowmon.GetClassifier ( ));

std::map<FlowId, FlowMonitor::FlowStats> stats = monitor->GetFlowStats ();

for (std::map<FlowId, FlowMonitor::FlowStats>::const\_iterator i = stats.begin (); i != stats.end (); ++i)

{

Ipv4FlowClassifier::FiveTuple t = classifier->FindFlow (i->first);

std::cout<<" t.sourceAddress="<<t.sourceAddress<<" t.destinationAddress="<<t.destinationAddress<<" Source port:"<<t.sourcePort<<" " <<"Destination port:"<<t.destinationPort<<"\n";

if ((t.sourceAddress=="10.1.1.1" && t.destinationAddress == "10.1.1.2"))

{

std::cout << "Flow " << i->first << " (" << t.sourceAddress << " -> " << t.destinationAddress << ")\n";

std::cout << " Tx Bytes: " << i->second.txBytes << "\n";

std::cout << " Rx Bytes: " << i->second.rxBytes << "\n";

std::cout << " Tx Packets: " << i->second.txPackets << "\n";

std::cout << " Rx Packets: " << i->second.rxPackets << "\n";

std::cout << " Throughput: " << i->second.rxBytes \* 8.0 / 9.0 / 1000 / 1000 << " Mbps\n";

}

}

monitor->SerializeToXmlFile("lab-1.flowmon", true, true);

Simulator::Destroy ();

NS\_LOG\_INFO ("Done.");

}

1. **Create a topology as given below**

**10.1.1.0**

**n0 -------------- n1 n2 n3 n4**

**point-to-point | | | |**

**==============**

**LAN 10.1.2.0**

**Demonstrate the use of NetAnim.**

**SOLUTION:**

#include "ns3/core-module.h"

#include "ns3/network-module.h"

#include "ns3/csma-module.h"

#include "ns3/internet-module.h"

#include "ns3/point-to-point-module.h"

#include "ns3/applications-module.h"

#include "ns3/ipv4-global-routing-helper.h"

#include "ns3/netanim-module.h"

#include "ns3/mobility-module.h"

#include "ns3/animation-interface.h"

// Default Network Topology

//

// 10.1.1.0

// n0 -------------- n1 n2 n3 n4

// point-to-point | | | |

// ================

// LAN 10.1.2.0

using namespace ns3;

NS\_LOG\_COMPONENT\_DEFINE ("SecondScriptExample");

int

main (int argc, char \*argv[])

{

bool verbose = true;

uint32\_t nCsma = 3,np2p=2;

CommandLine cmd;

cmd.AddValue ("nCsma", "Number of \"extra\" CSMA nodes/devices", nCsma);

cmd.AddValue ("verbose", "Tell echo applications to log if true", verbose);

cmd.Parse (argc,argv);

if (verbose)

{

LogComponentEnable ("UdpEchoClientApplication", LOG\_LEVEL\_INFO);

LogComponentEnable ("UdpEchoServerApplication", LOG\_LEVEL\_INFO);

}

nCsma = nCsma == 0 ? 1 : nCsma;

NodeContainer p2pNodes;

p2pNodes.Create (np2p);

NodeContainer csmaNodes;

csmaNodes.Add (p2pNodes.Get (1));

csmaNodes.Create (nCsma);

PointToPointHelper pointToPoint;

pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));

pointToPoint.SetChannelAttribute ("Delay", StringValue ("2ms"));

NetDeviceContainer p2pDevices;

p2pDevices = pointToPoint.Install (p2pNodes);

CsmaHelper csma;

csma.SetChannelAttribute ("DataRate", StringValue ("100Mbps"));

csma.SetChannelAttribute ("Delay", TimeValue (NanoSeconds (6560)));

NetDeviceContainer csmaDevices;

csmaDevices = csma.Install (csmaNodes);

InternetStackHelper stack;

stack.Install (p2pNodes.Get (0));

stack.Install (csmaNodes);

Ipv4AddressHelper address;

address.SetBase ("10.1.1.0", "255.255.255.0");

Ipv4InterfaceContainer p2pInterfaces;

p2pInterfaces = address.Assign (p2pDevices);

address.SetBase ("10.1.2.0", "255.255.255.0");

Ipv4InterfaceContainer csmaInterfaces;

csmaInterfaces = address.Assign (csmaDevices);

UdpEchoServerHelper echoServer (9);

ApplicationContainer serverApps = echoServer.Install (csmaNodes.Get (nCsma));

serverApps.Start (Seconds (1.0));

serverApps.Stop (Seconds (10.0));

UdpEchoClientHelper echoClient (csmaInterfaces.GetAddress (nCsma), 9);

echoClient.SetAttribute ("MaxPackets", UintegerValue (1));

echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.0)));

echoClient.SetAttribute ("PacketSize", UintegerValue (1024));

ApplicationContainer clientApps = echoClient.Install (p2pNodes.Get (0));

clientApps.Start (Seconds (2.0));

clientApps.Stop (Seconds (10.0));

Ipv4GlobalRoutingHelper::PopulateRoutingTables ();

pointToPoint.EnablePcapAll ("second");

csma.EnablePcap ("second", csmaDevices.Get (1), true);

AnimationInterface anim ("anim\_sec.xml");

anim.SetConstantPosition (p2pNodes.Get (0), 0.5, 0.5);

anim.SetConstantPosition (csmaNodes.Get (0), 1.0, 1.0);

anim.SetConstantPosition (csmaNodes.Get (1), 1.5,1.25);

anim.SetConstantPosition (csmaNodes.Get (2), 2.0,1.5);

anim.SetConstantPosition (csmaNodes.Get (3), 2.5,1.75);

Simulator::Run ();

Simulator::Destroy ();

return 0;

}

**CYCLE 3**

1. Using TCP/IP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present.

**SOLUTION:**

**Client.py**

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:', filecontents)

clientSocket.close()

**Server.py**

from socket import \*

serverName=”127.0.0.1”

serverPort = 12000

serverSocket = socket(AF\_INET,SOCK\_STREAM)

serverSocket.bind((serverName,serverPort))

serverSocket.listen(1)

print ("The server is ready to receive")

while 1:

connectionSocket, addr = serverSocket.accept()

sentence = connectionSocket.recv(1024).decode()

file=open(sentence,"r")

l=file.read(1024)

connectionSocket.send(l.encode())

file.close()

connectionSocket.close()

1. **Using UDP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present.**

**SOLUTION:**

**ClientUDP.py**

from socket import \*

serverName = "127.0.0.1"

serverPort = 12000

clientSocket = socket(AF\_INET, SOCK\_DGRAM)

sentence = input("Enter file name")

clientSocket.sendto(bytes(sentence,"utf-8"),(serverName, serverPort))

filecontents,serverAddress = clientSocket.recvfrom(2048)

print ('From Server:', filecontents)

clientSocket.close()

**ServerUDP.py**

from socket import \*

serverPort = 12000

serverSocket = socket(AF\_INET, SOCK\_DGRAM)

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

print ("The server is ready to receive")

while 1:

sentence,clientAddress = serverSocket.recvfrom(2048)

file=open(sentence,"r")

l=file.read(2048)

serverSocket.sendto(bytes(l,"utf-8"),clientAddress)

print("sent back to client",l)

file.close()

1. **Implement the above program using as message queues or FIFO as IPC channels.**

**SOLUTION:**

/\*Server\*/

#include<stdio.h>

#include<unistd.h>

#include<sys/stat.h>

#include<fcntl.h>

#include<string.h>

#define FIFO1 "fifo1"

#define FIFO2 "fifo2"

#define PERMS 0666

char fname[256];

int main( ) {

int readfd, writefd, fd;

ssize\_t n;

char buff[512];

if (mkfifo(FIFO1, PERMS)<0)

printf("Cant Create FIFO Files\n");

if (mkfifo(FIFO2, PERMS)<0)

printf("Cant Create FIFO Files\n");

printf("Waiting for connection Request..\n");

readfd =open(FIFO1, O\_RDONLY, 0);

writefd=open(FIFO2, O\_WRONLY, 0);

printf("Connection Established..\n");

read(readfd, fname, 255);

printf("Client has requested file %s\n",fname);

if ((fd=open(fname,O\_RDWR))<0) {

strcpy(buff,"File does not exist..\n");

write(writefd, buff, strlen(buff));

} else {

while((n=read(fd, buff,512))>0)

write(writefd, buff, n);

}

close(readfd); unlink(FIFO1);

close(writefd); unlink(FIFO2);

}

/\*Client\*/

#include<stdio.h>

#include<unistd.h>

#include<sys/stat.h>

#include<fcntl.h>

#define FIFO1 "fifo1"

#define FIFO2 "fifo2"

#define PERMS 0666

char fname[256];

int main()

{

ssize\_t n;

char buff[512];

int readfd,writefd;

printf("Trying to Connect to Server..\n");

writefd = open(FIFO1, O\_WRONLY, 0);

readfd = open(FIFO2, O\_RDONLY, 0);

printf("Connected..\n");

printf("Enter the filename to request from server: ");

scanf("%s",fname);

write(writefd, fname, strlen(fname));

printf("Waiting for Server to reply..\n");

while((n=read(readfd,buff,512))>0)

write(1,buff,n);

close(readfd);

close(writefd);

return 0;

}

Output (Server)

[root@localhost CN Lab] ./s.o

Waiting for connection Request..

Connection Established..

Client has requested file alpha

[root@localhost CN Lab]

Output (Client)

[root@localhost CN Lab] ./c.o

Trying to Connect to Server..

Connected..

Enter the filename to request from server: alpha

Waiting for Server to reply..

This a demo of client server using Sockets

Just for trial.

Now End of file

1. **Write a program for congestion control using Leaky bucket algorithm.**

#include<iostream.h>

#include<dos.h>

#include<stdlib.h>

#define bucketSize 512

void bktInput(int a,int b) {

if(a>bucketSize)

cout<<"\n\t\tBucket overflow";

else {

delay(500);

while(a>b){

cout<<"\n\t\t"<<b<<" bytes outputted.";

a-=b;

delay(500);

}

if (a>0) cout<<"\n\t\tLast "<<a<<" bytes sent\t";

cout<<"\n\t\tBucket output successful";

}

}

void main( ) {

int op, pktSize;

randomize();

cout<<"Enter output rate : "; cin>>op;

for(int i=1;i<=5;i++){

delay(random(1000));

pktSize=random(1000);

cout<<"\nPacket no "<<i<<"\tPacket size = "<<pktSize;

bktInput(pktSize,op);

}

}

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

Last 95 bytes sent

Packet no 4 Packet size = 949

Bucket overflow

1. **Write a program for distance vector algorithm to find suitable path for transmission.**

**SOLUTION:**

#include <conio.h>

#include <iostream.h>

#define MAX 10

int n;

class router {

char adj\_new[MAX], adj\_old[MAX];

int table\_new[MAX], table\_old[MAX];

public:

router( ){

for(int i=0;i<MAX;i++) table\_old[i]=table\_new[i]=99;

}

void copy( ){

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

adj\_old[i] =adj\_new[i];

table\_old[i]=table\_new[i];

}

}

int equal( ) {

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

if(table\_old[i]!=table\_new[i]||adj\_new[i]!=adj\_old[i])return 0;

return 1;

}

void input(int j) {

cout<<"Enter 1 if the corresponding router is adjacent to router"

<<(char)('A'+j)<<" else enter 99: "<<endl<<" ";

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

if(i!=j) cout<<(char)('A'+i)<<" ";

cout<<"\nEnter matrix:";

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

if(i==j)

table\_new[i]=0;

else

cin>>table\_new[i];

adj\_new[i]= (char)('A'+i);

}

cout<<endl;

}

void display(){

cout<<"\nDestination Router: ";

for(int i=0;i<n;i++) cout<<(char)('A'+i)<<" ";

cout<<"\nOutgoing Line: ";

for(i=0;i<n;i++) cout<<adj\_new[i]<<" ";

cout<<"\nHop Count: ";

for(i=0;i<n;i++) cout<<table\_new[i]<<" ";

}

void build(int j) {

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

for(int k=0;(i!=j)&&(k<n);k++)

if(table\_old[i]!=99)

if((table\_new[i]+r[i].table\_new[k])<table\_new[k]) {

table\_new[k]=table\_new[i]+r[i].table\_new[k];

adj\_new[k]=(char)('A'+i);

}

}

} r[10];

void build\_table( ) {

int i=0, j=0;

while(i!=n) {

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

r[i].copy();

r[i].build(i);

}

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

if(!r[i].equal()) {

j=i;

break;

}

}

}

void main() {

clrscr();

cout<<"Enter the number the routers(<"<<MAX<<"): "; cin>>n;

for(int i=0;i<n;i++) r[i].input(i);

build\_table();

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

cout<<"Router Table entries for router "<<(char)('A'+i)<<":-";

r[i].display();

cout<<endl<<endl;

}

getch();

}

**Output**

Enter the number the routers: 5

Enter 1 if the corresponding is adjacent to router A else enter 99:

B C D E

Enter matrix:1 1 99 99

Enter 1 if the corresponding is adjacent to router B else enter 99:

A C D E

Enter matrix:1 99 99 99

Enter 1 if the corresponding is adjacent to router C else enter 99:

A B D E

Enter matrix:1 99 1 1

Enter 1 if the corresponding is adjacent to router D else enter 99:

A B C E

Enter matrix:99 99 1 99

Enter 1 if the corresponding is adjacent to router E else enter 99:

A B C D

Enter matrix:99 99 1 99

Router Table entries for router A

Destination Router: A B C D E

Outgoing Line: A B C CC

Hop Count: 0 1 1 2 2

Router Table entries for router B

Destination Router: A B C D E

Outgoing Line: A B A AA

Hop Count: 1 0 2 3 3

Router Table entries for router C

Destination Router: A B C D E

Outgoing Line: A A C D E

Hop Count: 1 2 0 1 1

Router Table entries for router D

Destination Router: A B C D E

Outgoing Line: C CC D C

Hop Count: 2 3 1 0 2

Router Table entries for router E

Destination Router: A B C D E

Outgoing Line: C CCC E

Hop Count: 2 3 1 2 0

1. **Implement Dijkstra’s algorithm to compute the shortest path through a graph.**

**SOLUTION:**

|  |
| --- |
| #include<stdio.h> |
| #include<conio.h> |

|  |  |
| --- | --- |
| #include<process.h> | |
| #include<string.h> |

|  |  |
| --- | --- |
| #include<math.h> | |
| #define IN 99 |

|  |
| --- |
| #define N 6 |
| int dijkstra(int cost[][N], int source, int target); | |

|  |  |
| --- | --- |
| int dijsktra(int cost[][N],int source,int target) | |
| { |

|  |  |
| --- | --- |
| int dist[N],prev[N],selected[N]={0},i,m,min,start,d,j; | |
| char path[N]; |

|  |  |
| --- | --- |
| for(i=1;i< N;i++) | |
| { |

|  |
| --- |
| dist[i] = IN; |
| prev[i] = -1; |

|  |
| --- |
| } |
| start = source; | |

|  |  |
| --- | --- |
| selected[start]=1; | |
| dist[start] = 0; |

|  |  |
| --- | --- |
| while(selected[target] ==0) | |
| { |

|  |  |
| --- | --- |
| min = IN; | |
| m = 0; |

|  |  |
| --- | --- |
| for(i=1;i< N;i++) | |
| { |

|  |  |
| --- | --- |
| d = dist[start] +cost[start][i]; | |
| if(d< dist[i]&&selected[i]==0) |

|  |
| --- |
| { |
| dist[i] = d; | |

|  |  |
| --- | --- |
| prev[i] = start; | |
| } |

|  |  |
| --- | --- |
| if(min>dist[i] && selected[i]==0) | |
| { |

|  |  |
| --- | --- |
| min = dist[i]; | |
| m = i; |

|  |  |
| --- | --- |
| } | |
| } |

|  |
| --- |
| start = m; |
| selected[start] = 1; | |

|  |
| --- |
| } |
| start = target; | |

|  |
| --- |
| j = 0; |
| while(start != -1) | |

|  |
| --- |
| { |
| path[j++] = start+65; | |

|  |  |
| --- | --- |
| start = prev[start]; | |
| } |

|  |
| --- |
| path[j]='\0'; |
| strrev(path); |

|  |
| --- |
| printf("%s", path); |
| return dist[target]; | |

|  |
| --- |
| } |
| int main() | |

|  |
| --- |
| { |
| int cost[N][N],i,j,w,ch,co; | |

|  |
| --- |
| int source, target,x,y; |
| printf("\tShortest Path Algorithm(DIJKSRTRA's ALGORITHM\n\n"); | |

|  |
| --- |
| for(i=1;i< N;i++) |
| for(j=1;j< N;j++) |

|  |
| --- |
| cost[i][j] = IN; |
| for(x=1;x< N;x++) | |

|  |
| --- |
| { |
| for(y=x+1;y< N;y++) | |

|  |
| --- |
| { |
| printf("Enter the weight of the path between node %d and %d: ",x,y); | |

|  |
| --- |
| scanf("%d",&w); |
| cost [x][y] = cost[y][x] = w; | |

|  |
| --- |
| } |
| printf("\n"); | |

|  |
| --- |
| } |
| printf("\nEnter The Source:"); | |

|  |
| --- |
| scanf("%d", &source); |
| printf("\nEnter The target"); | |

|  |
| --- |
| scanf("%d", &target); |
| co = dijsktra(cost,source,target); | |

|  |  |
| --- | --- |
| printf("\nShortest Path: %d",co); | |
| } |