**EX NO:2 IMPLEMENTATION OF DISTANCE VECTOR ROUTING**

**ALGORITHM**

**AIM:**

To implement the distance vector routing algorithm.

**APPARATUS REQUIRED:**

\* Pentium 4 PC – 1

\* Cisco Packet Tracer software.

**PRINCIPLE:**

It is under dynamic routing algorithm. This algorithm operates by having each route maintains a table giving the least known distance to reach destination and include line in used to get these. These are updated by changing information with neighbour. This is called “Bell mann ford algorithm” and “fod fick” algorithm.

**PROCEDURE:**

\* Open the Cisco Packet Tracer software.

\* Add the router and PCs accourding to our design.

\* Configure all the routers and PCs.

\* Trace the destination in PC’s command prompt.

\* Verify the output.

**Configuring routers:**

**Router 0**:

Router>en

Router#config t

Router(config)#int fa0/0

Router(config-if)#ip add 10.0.0.2 255.0.0.0

Router(config-if)#no shut

Router(config-if)#exit

Router(config)#int s0/0/0

Router(config-if)#ip add 20.0.0.1 255.0.0.0

Router(config-if)#clock rate 64000

Router(config-if)#no shut

Router(config-if)#exit

Router(config)#int s0/0/1

Router(config-if)#ip add 50.0.0.2 255.0.0.0

Router(config-if)#clock rate 64000

Router(config-if)#no shut

Router(config-if)#exit

Router(config)#router rip

Router(config-router)#network 10.0.0.0

Router(config-router)#network 20.0.0.0

Router(config-router)#network 50.0.0.0

Router(config-router)#exitRouter(config)#exit

**Router 1**:

Router>en

Router#config t

Router(config)#int s0/0/0

Router(config-if)#ip add 20.0.0.2 255.0.0.0

Router(config-if)#clock rate 64000

Router(config-if)#no shut

Router(config-if)#exit

Router(config)#int s0/0/1

Router(config-if)#ip add 30.0.0.1 255.0.0.0

Router(config-if)#clock rate 64000

Router(config-if)#no shut

Router(config-if)#exit

Router(config)#router rip

Router(config-router)#network 20.0.0.0

Router(config-router)#network 30.0.0.0

Router(config-router)#exit

Router(config)#exit

**Router 2** :

Router>en

Router#config t

Router(config)#int fa0/0

Router(config-if)#ip add 40.0.0.1 255.0.0.0

Router(config-if)#no shut

Router(config-if)#exit

Router(config)#int s0/0/0

Router(config-if)#ip add 30.0.0.2 255.0.0.0

Router(config-if)#clock rate 64000

Router(config-if)#no shut

Router(config-if)#exit

Router(config)#int s0/0/1

Router(config-if)#ip add 50.0.0.1 255.0.0.0

Router(config-if)#clock rate 64000

Router(config-if)#no shut

Router(config-if)#exit

Router(config)#router rip

Router(config-router)#network 40.0.0.0

Router(config-router)#network 30.0.0.0

Router(config-router)#network 50.0.0.0

Router(config-router)#exit

Router(config)#exit

Router#

**Configuring PCs:**

**PC0** 🡺 Desktop 🡺 IP Configuration

IP Address : 10.0.0.1

Subnet Mask : 255.0.0.0

Default Gateway : 10.0.0.2

**PC1** 🡺 Desktop 🡺 IP Configuration

IP Address : 40.0.0.2

Subnet Mask : 255.0.0.0

Default Gateway : 40.0.0.1

**Flow Diagram :**

R2

R0

s0/0 /0

20.0.0.2

s0/0 /1

30.0.0.1

s0/0 /1

50.0.0.2

s0/0 /1

50.0.0.1

fa0/0 10.0.0.2

40.0.0.2

10.0.0.1

fa0/0

40.0.0.1

s0/0 /0

30.0.0.2

s0/0 /0

20.0.0.1

Pc0

Pc1

R1

**Router Table** :

**output:**

|  |  |
| --- | --- |
| **PC0:** | **PC1:** |
| **Default:**  PC>tracert 40.0.0.2  Tracing route to 40.0.0.2 over a maximum of 30 hops:  1 31 ms 31 ms 31 ms 10.0.0.2  2 62 ms 62 ms 62 ms 50.0.0.1  3 66 ms 78 ms 94 ms 40.0.0.2  Trace complete. | **Default:**  PC>tracert 10.0.0.1  Tracing route to 10.0.0.1 over a maximum of 30 hops:  1 31 ms 32 ms 16 ms 40.0.0.1  2 63 ms 47 ms 63 ms 50.0.0.2  3 79 ms 93 ms 93 ms 10.0.0.1  Trace complete. |

**RESULT:**

Thus the distance vector routing algorithm was implemented and the output was verified.

**EX NO:3 IMPLEMENTATION OF LINK STATE ROUTING ALGORITHM**

**AIM:**

To implement link state routing algorithm.

**APPARATUS REQUIRED:**

\* Pentium 4 PC – 2

\* Cisco Packet Tracer software

**PRINCIPLE:**

Link state routing works on the following principle.

\* Discover the neighbour and keep their network address.

\* Measure the delay or cost to each of its neighbour.

\* Construct a packet telling all it has just learned.

\* Send the packet to all router.

\* Compute the shortest path to every router.

**PROCEDURE:**

\* Open the Cisco Packet Tracer software.

\* Add the router and PCs according to our design.

\* Configure all the routers and PCs.

\* Trace the destination in PC’s command prompt.

\* Verify the output.

**Configuring Routes:**

**router 0: 🡺 CLI**

Router>en

Router#config t

Router(config)#int fa0/0

Router(config-if)#ip add 10.0.0.2 255.0.0.0

Router(config-if)#no shut

Router(config-if)#exit

Router(config)#int s0/0/0

Router(config-if)#ip add 20.0.0.1 255.0.0.0

Router(config-if)#clock rate 64000

Router(config-if)#no shut

Router(config-if)#exit

Router(config)#int s0/0/1

Router(config-if)#ip add 50.0.0.2 255.0.0.0

Router(config-if)#clock rate 64000

Router(config-if)#no shut

Router(config-if)#router ospf 100

Router(config-router)#network 10.0.0.0 0.255.255.255 area 0

Router(config-router)#network 20.0.0.0 0.255.255.255 area 0

Router(config-router)#network 50.0.0.0 0.255.255.255 area 0

Router(config-router)#exit

Router(config)#

Router#

**router 1: 🡺 CLI**

Router>en

Router#config t

Router(config)#int s0/0/0

Router(config-if)#ip add 20.0.0.2 255.0.0.0

Router(config-if)#clock rate 64000

Router(config-if)#no shut

Router(config-if)#exit

Router(config)#int s0/0/1

Router(config-if)#ip add 30.0.0.1 255.0.0.0

Router(config-if)#clock rate 64000

Router(config-if)#no shut

Router(config)#router ospf 100

Router(config-router)#network 20.0.0.0 0.255.255.255 area 0

Router(config-router)#network 30.0.0.0 0.255.255.255 area 0

Router(config-router)#exit

Router(config)#exit

Router#

**Router 2: 🡺 CLI**

Router>en

Router#config t

Router(config)#int fa0/0

Router(config-if)#ip add 40.0.0.1 255.0.0.0

Router(config-if)#no shut

Router(config-if)#exit

Router(config)#int s0/0/0

Router(config-if)#ip add 30.0.0.2 255.0.0.0

Router(config-if)#clock rate 64000

Router(config-if)#no shut

Router(config-if)#exit

Router(config)#int s0/0/1

Router(config-if)#ip add 50.0.0.1 255.0.0.0

Router(config-if)#clock rate 64000

Router(config-if)#no shut

Router(config-router)#router ospf 100

Router(config-router)#network 40.0.0.0 0.255.255.255 area 0

Router(config-router)#network 30.0.0.0 0.255.255.255 area 0

Router(config-router)#network 50.0.0.0 0.255.255.255 area 0

Router(config-router)#exit

Router(config)#exit

Router#

**Configuring PCs:**

**PC0** 🡺 Desktop 🡺 IP Configuration

IP Address : 10.0.0.1

Subnet Mask : 255.0.0.0

Default Gateway : 10.0.0.2

**PC1** 🡺 Desktop 🡺 IP Configuration

IP Address : 40.0.0.2

Subnet Mask : 255.0.0.0

Default Gateway : 40.0.0.1

**To Change cost:**

(cost to be changed in both terminal of the path)

**Router 0 🡺 CLI** (at 50.0.0.2 terminal)

Router>en

Router#config t

Router(config)#int s0/0/01 (Respective terminal number)

Router(config-if)#ip ospf cost 200 (user define cost)

Router(config-if)#exit

Router(config)#

Router#

**Router 2 🡺 CLI** (at 50.0.0.1 terminal)

Router>en

Router#config t

Router(config)#int s0/0/1

Router(config-if)#ip ospf cost 200

Router(config-if)#exit

Router(config)#^

Router#

Abbreviation

Ospf = Open Shortest Path First

RIP: Routing Information Protocol

**Flow Diagram :**

R2

R0

s0/0 /0

20.0.0.2

s0/0 /1

30.0.0.1

s0/0 /1

50.0.0.2

s0/0 /1

50.0.0.1

fa0/0 10.0.0.2

40.0.0.2

10.0.0.1

fa0/0

40.0.0.1

s0/0 /0

30.0.0.2

s0/0 /0

20.0.0.1

Pc0

Pc1

R1

**Router Table** :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Route 0** | | **Route 2** | | **Route 3** | |
| **Interface** | **Ip add** | **Interface** | **Ip add** | **Interface** | **Ip add** |
| fa0 | 10.0.0.2 | s0 | 20.0.0.2 | fa0 | 40.0.0.1 |
| s0 | 20.0.0.1 | s1 | 30.0.0.1 | s0 | 30.0.0.2 |
| s1 | 50.0.0.2 |  |  | s1 | 50.0.0.1 |
| Pc0 | 10.0.0.1 |  |  | Pc1 | 40.0.0.2 |

**output:**

|  |  |
| --- | --- |
| **PC0:** | **PC1:** |
| **Default:**  PC>tracert 40.0.0.2  Tracing route to 40.0.0.2 over a maximum of 30 hops:  1 31 ms 31 ms 31 ms 10.0.0.2  2 62 ms 62 ms 62 ms 50.0.0.1  3 66 ms 78 ms 94 ms 40.0.0.2  Trace complete. | **Default:**  PC>tracert 10.0.0.1  Tracing route to 10.0.0.1 over a maximum of 30 hops:  1 31 ms 32 ms 16 ms 40.0.0.1  2 63 ms 47 ms 63 ms 50.0.0.2  3 79 ms 93 ms 93 ms 10.0.0.1  Trace complete. |
| **When path2 cost changed to 200:**  PC>tracert 40.0.0.2  Tracing route to 40.0.0.2 over a maximum of 30 hops:  1 47 ms 31 ms 31 ms 10.0.0.2  2 62 ms 62 ms 62 ms 20.0.0.2  3 94 ms 93 ms 93 ms 30.0.0.2  4 125 ms 125 ms 125 ms 40.0.0.2  Trace complete. | **When path2 cost changed to 200:**  PC>tracert 10.0.0.1  Tracing route to 10.0.0.1 over a maximum of 30 hops:  1 17 ms 31 ms 31 ms 40.0.0.1  2 63 ms 63 ms 63 ms 30.0.0.1  3 66 ms 94 ms 78 ms 20.0.0.1  4 125 ms 112 ms 125 ms 10.0.0.1  Trace complete. |

**RESULT**:

Thus the link state algorithm was implemented and the output was verified.