EXPERIMENT NO -8

AIM: IMPLEMENTATION OF DVR/LSR IN NS2.

DESCRIPTION:

In Distance Vector (DV), each node sends periodic route updates for every 2 seconds. Apart from the periodic updates, each node/agent sends triggered updates as a result of changes in the forwarding table in the node if any. This occurs either due to changes in the network topology or the node received a route update and as a result, it composes a fresh route.

DEFINITION:

A **distance**-**vector routing** (DVR) protocol requires that a **router** inform its neighbors of topology changes periodically. Historically known as the old ARPANET **routing** algorithm (or known as Bellman-Ford algorithm).  **Distances** based on a chosen metric, are computed using information from the neighbors' **distance vectors**.

**Distance Vector Algorithm –**

1. A router transmits its distance vector to each of its neighbors in a routing packet.
2. Each router receives and saves the most recently received distance vector from each of its neighbors.
3. A router recalculates its distance vector when:
   * It receives a distance vector from a neighbor containing different information than before.
   * It discovers that a link to a neighbor has gone down.

The DV calculation is based on minimizing the cost to each destination.

LINK STATE ROUTING (LSR)

**Link State Routing**. **Link state routing** is a technique in which each **router** shares the knowledge of its neighborhood with every other **router** in the internetwork.

The basic concept of **link**-**state routing** is that every node constructs a map of the connectivity to the network, in the form of a graph, showing which nodes are connected to which other nodes. Each node then independently calculates the next best logical path from it to every possible destination in the network.

**Features of link state routing protocols –**

* **Link state packet –** A small packet that contains routing information.
* **Link state database –** A collection information gathered from link state packet.
* **Shortest path first algorithm (Dijkstra algorithm) –** A calculation performed on the database results into shortest path
* **Routing table –** A list of known paths and interfaces.

There are three major protocols for unicast routing:

1. Distance Vector Routing
2. Link State Routing
3. Path-Vector Routing

PROGRAM:

set ns [new Simulator]  
  
set nf [open out.nam w]  
$ns namtrace-all $nf  
  
set tr [open out.tr w]  
$ns trace-all $tr  
  
proc finish {} {  
        global nf ns tr  
        $ns flush-trace  
        close $tr  
        exec nam out.nam &  
        exit 0  
        }  
  
set n0 [$ns node]  
set n1 [$ns node]  
set n2 [$ns node]  
set n3 [$ns node]  
  
$ns duplex-link $n0 $n1 10Mb 10ms DropTail  
$ns duplex-link $n1 $n3 10Mb 10ms DropTail  
$ns duplex-link $n2 $n1 10Mb 10ms DropTail  
  
$ns duplex-link-op $n0 $n1 orient right-down  
$ns duplex-link-op $n1 $n3 orient right  
$ns duplex-link-op $n2 $n1 orient right-up  
  
set tcp [new Agent/TCP]  
$ns attach-agent $n0 $tcp  
  
set ftp [new Application/FTP]  
$ftp attach-agent $tcp  
  
set sink [new Agent/TCPSink]  
$ns attach-agent $n3 $sink  
  
set udp [new Agent/UDP]  
$ns attach-agent $n2 $udp  
  
set cbr [new Application/Traffic/CBR]  
$cbr attach-agent $udp  
  
set null [new Agent/Null]  
$ns attach-agent $n3 $null  
  
$ns connect $tcp $sink  
$ns connect $udp $null  
  
$ns rtmodel-at 1.0 down $n1 $n3  
$ns rtmodel-at 2.0 up $n1 $n3  
  
$ns rtproto DV  
  
$ns at 0.0 "$ftp start"  
$ns at 0.0 "$cbr start"  
  
$ns at 5.0 "finish"  
  
$ns run

OUTPUT:

