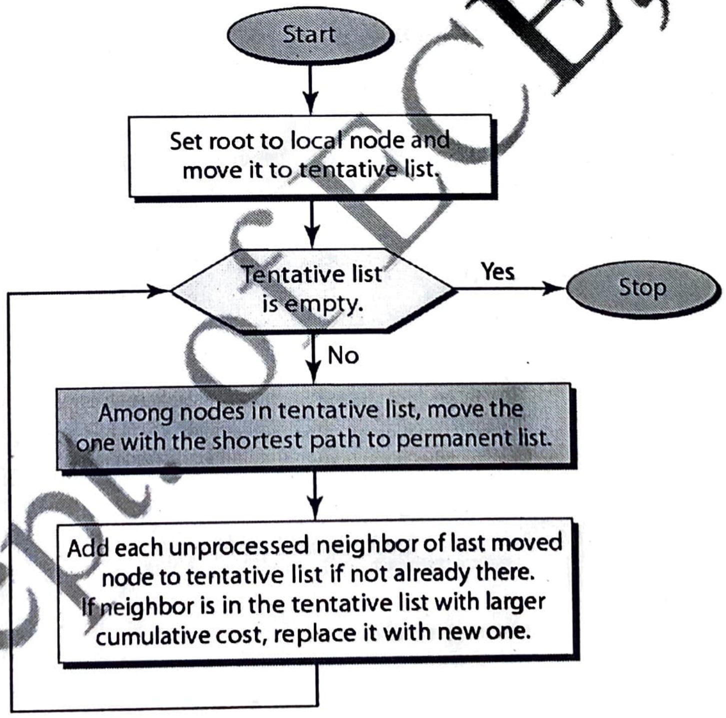
Experiment -6

Implementation of Link state routing algorithm

Pre-requisites:

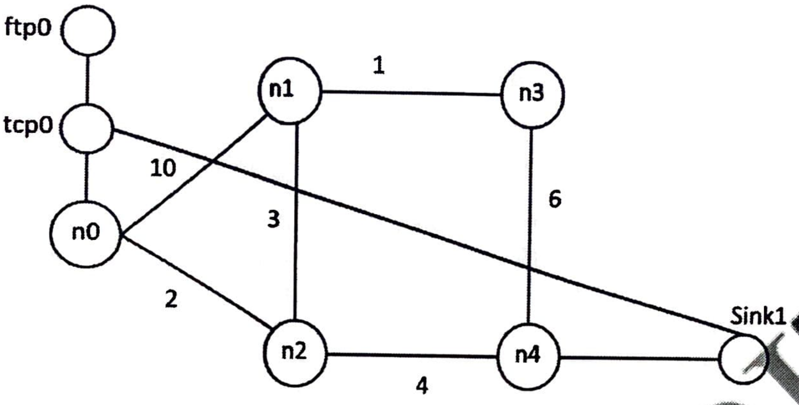
Link State Routing Algorithm: A routing algorithm that directly allows creating least-cost trees and forwarding tables is called link-state (LS) routing. This method uses the term link-state to define the characteristic of a link (an edge) that represents a network in the internet. To create a least-cost tree with this method, each node needs to have a complete map of the network, which means it needs to know the state of each link. The collection of states for all links is called the link-state database (LSDB).There is only one LSDB for the whole internet. Now the question is how each node can create this LSDB that contains information about the whole internet? This can be done by a process called flooding. Each node can send some greeting messages to all its immediate neighbors (those nodes to which it is connected directly) to collect two pieces of information for each neighboring node. The identity of the node and the cost of the link. To create a least-cost tree for itself, using the shared LSDB, each node needs to run the famous Dijkstra Algorithm


Start
Set root to local node and move it to tentative list.
Tentative list is empty.
Yes
Stop
No
Among nodes in tentative list, move the one with the shortest path to permanent list.
Add each unprocessed neighbor of last moved node to tentative list if not already there. If neighbor is in the tentative list with larger cumulative cost, replace it with new one.

Requirements for Network Setup:

- 5 nodes
- Point to point network
- TCP agent and FTP application

Network Diagram:



TCL Script:

```
#===================================
#   Simulation parameters setup
#===================================
setval(stop)   10.0                 ;# time of simulation end

#===================================
#      Initialization
#===================================
#Create a ns simulator
set ns [new Simulator]

#Open the NS trace file
settracefile [open lab6.tr w]
$ns trace-all $tracefile

#Open the NAM trace file
setnamfile [open lab6.nam w]
$ns namtrace-all $namfile

#===================================
#      Nodes Definition
#===================================
#Create 5 nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]

#===================================
#      Links Definition
#===================================
#Createlinks between nodes
$ns duplex-link $n0 $n1 100.0Mb 10ms DropTail
$ns queue-limit $n0 $n1 50
$ns duplex-link $n1 $n2 100.0Mb 10ms DropTail
```

```
$ns queue-limit $n1 $n2 50
$ns duplex-link $n2 $n3 100.0Mb 10ms DropTail
$ns queue-limit $n2 $n3 50
$ns duplex-link $n3 $n4 100.0Mb 10ms DropTail
$ns queue-limit $n3 $n4 50
$ns duplex-link $n4 $n0 100.0Mb 10ms DropTail
$ns queue-limit $n4 $n0 50
$ns duplex-link $n4 $n1 100.0Mb 10ms DropTail
$ns queue-limit $n4 $n1 50
```

Add this Lines →

```
$ns Cost $no $n1 10
$ns Cost $no $n2 2
$ns Cost $n1 $n3 1
$ns cost $n1 $n0 3
$ns cost $n2 $n4 4
$ns cost $n3 $n4 6
```

```
#Give node position (for NAM)
$ns duplex-link-op $n0 $n1 orient right-up
$ns duplex-link-op $n1 $n2 orient right
$ns duplex-link-op $n2 $n3 orient left-down
$ns duplex-link-op $n3 $n4 orient left

$ns duplex-link-op $n4 $n0 orient left-up
$ns duplex-link-op $n4 $n1 orient left-up

#===================================
#     Agents Definition
#===================================
#Setup a TCP connection
set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
set sink1 [new Agent/TCPSink]
$ns attach-agent $n3 $sink1
$ns connect $tcp0 $sink1
$tcp0 set packetSize_ 1500


#===================================
#     Applications Definition
#===================================
#Setup a FTP Application over TCP connection
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
$ns at 1.0 "$ftp0 start"
$ns at 4.0 "$ftp0 stop"

$ns rtproto LS                      ## to be added manually ##


#===================================
#     Termination
#===================================
#Define a 'finish' procedure
proc finish {} {
global ns tracefilenamfile
   $ns flush-trace
close $tracefile
close $namfile
execnam lab6.nam &
exit 0
}
```

```
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "finish"
$ns at $val(stop) "puts \"done\" ; $ns halt"
$ns run
```

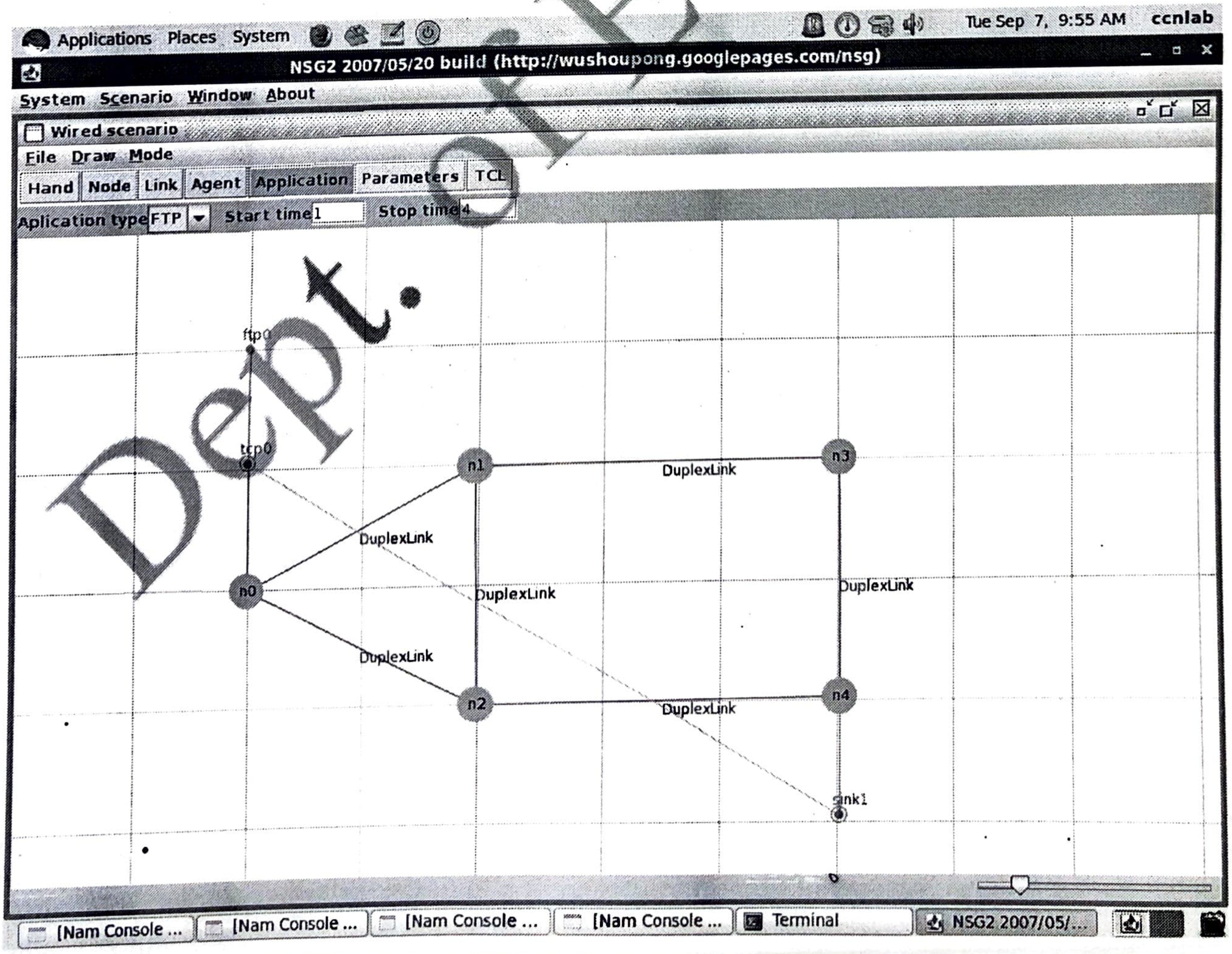
Execution Steps:

1. Create TCL code, save as eg: lab6.tcl
2. Run the network simulator in the terminal: ns lab5.tcl.
3. Observe the output in nam window and tracefile.
4. Tabulate the readings

Working:

Origin/Path	A	B	C	D	E
	∞	∞	∞	∞	∞
{A}	-	10	2 ✓	∞	∞
{A,C}	-	5 ✓	2	∞	6
{A,C,B}	-	5	2	6 ✓	6
{A,C,B,D}	-	5	2	6	6 ✓
{A,C,B,D,E}	-	5	2	6	6

Network Window:



NSG2 2007/05/20 build (http://wushoupong.googlepages.com/nsg)

Wired scenario

Aplication type FTP   Start time 1   Stop time 4