**Lab Experiment 2 :**

Implement simple ESS and with transmitting nodes in wire -less LAN by simulation and determine the performance with respect to transmission of packets.

**Topology**

****

Code-

**#create Simulator class**

set ns [new Simulator]

**#open trace file**

set nt [open lab2.tr w]

$ns trace-all $nt

**#create Topography object**

set topo [new Topography]

**#define grid size**

$topo load\_flatgrid 1000 1000

**#open namtrace file**

set nf [open lab2.nam w]

$ns namtrace-all-wireless $nf 1000 1000

**#specify node configuration**

$ns node-config -adhocRouting DSDV \

-llType LL \

-macType Mac/802\_11 \

-ifqType Queue/DropTail \

-ifqLen 20 \

-phyType Phy/WirelessPhy \

-channelType Channel/WirelessChannel \

-propType Propagation/TwoRayGround \

-antType Antenna/OmniAntenna \

-topoInstance $topo \

-agentTrace ON \

-routerTrace ON

**#create a General Operation Director(god) object that stores the total number of mobile nodes.**

create-god 4

**#create nodes and label them**

set n0 [$ns node]

set n1 [$ns node]

set n2 [$ns node]

set n3 [$ns node]

$n0 label "tcp0"

$n1 label "sink0"

$n2 label "bs1"

$n3 label "bs2"

**#give initial x, y, z coordinates to nodes**

$n0 set X\_ 110

$n0 set Y\_ 500

$n0 set Z\_ 0

$n1 set X\_ 600

$n1 set Y\_ 500

$n1 set Z\_ 0

$n2 set X\_ 300

$n2 set Y\_ 500

$n2 set Z\_ 0

$n3 set X\_ 450

$n3 set Y\_ 500

$n3 set Z\_ 0

**#attach agent and application to nodes and connect them**

set tcp0 [new Agent/TCP]

$ns attach-agent $n0 $tcp0

set ftp0 [new Application/FTP]

$ftp0 attach-agent $tcp0

set sink1 [new Agent/TCPSink]

$ns attach-agent $n1 $sink1

$ns connect $tcp0 $sink1

**#schedule the event**

$ns at 0.5 "$ftp0 start"

**#set up a destination for mobile nodes. They move to <x><y> coordinates at <s>m/s.**

$ns at 0.3 "$n0 setdest 110 500 10"

$ns at 0.3 "$n1 setdest 600 500 20"

$ns at 0.3 "$n2 setdest 300 500 30"

$ns at 0.3 "$n3 setdest 450 500 30"

$ns at 10.0 "$n0 setdest 100 550 5"

$ns at 10.0 "$n1 setdest 630 450 5"

$ns at 70.0 "$n0 setdest 170 680 5"

$ns at 70.0 "$n1 setdest 580 380 5"

$ns at 120.0 "$n0 setdest 140 720 5"

$ns at 135.0 "$n0 setdest 110 600 5"

$ns at 140.0 "$n1 setdest 600 550 5"

$ns at 155.0 "$n0 setdest 89 500 5"

$ns at 190.0 "$n0 setdest 100 440 5"

$ns at 210.0 "$n1 setdest 700 600 5"

$ns at 240.0 "$n1 setdest 650 500 5"

proc finish { } {

global ns nt nf

$ns flush-trace

exec nam lab42.nam &

close $nt

close $nf

exit 0

}

$ns at 400 "finish"

$ns run

**AWK file**

BEGIN{

PktsSent=0;

PktsRcvd=0;

PktsAtRTR=0;

}

{

if(($1=="s")&&($4=="RTR")&&($7=="tcp")) PktsAtRTR++;

if(($1=="s")&&($4=="AGT")&&($7=="tcp")) PktsSent++;

if(($1=="r")&&($4=="AGT")&&($7=="tcp")) PktsRcvd++;

}

END{

print " Number of Packets Sent :" PktsSent

print " Number of Packets Received :" PktsRcvd

print " Pacjet Delivery Ratio :" PktsRcvd/PktsSent\*100

print " Routing Load :" PktsAtRTR/PktsRcvd

}

**Output-**

$ns pgm6.tcl

$awk -f count.awk lab42.tr

Number of Packets Sent: 6819

Number of Packets Received: 6685

Packet Delivery Ratio: 98.0349

Routing Load: 1.02004

**Explanation:**

**set topo [new Topography]**

* The line set topo [new Topography] in an ns-2 script **creates a new instance of the Topography class and assigns it to the variable topo.**
* The Topography class is **used to model the spatial layout of nodes and their connectivity** in a network simulation.
* set topo: **Assigns the newly created Topography object to the variable topo**. This variable can then be used to configure or manipulate the topography of the network in the simulation.
* The Topography class in ns-2 is **used in conjunction with nodes, links, and agents to define the physical characteristics and layout of the simulated network.**

**$topo load\_flatgrid 1000 1000**

* The line $topo load\_flatgrid 1000 1000 in an ns-2 script uses the **load\_flatgrid method of the Topography object ($topo) to create a flat grid topology with specified dimensions**.
* $topo load\_flatgrid 1000 1000: This command **invokes** the load\_flatgrid method on the Topography object ($topo). The method takes two arguments, 1000 and 1000, which represent the **dimensions (width and height)** of the flat grid topology.
* The first argument (1000) specifies the width of the grid.
* The second argument (1000) specifies the height of the grid.
* This line essentially sets up a flat grid topology with a width of 1000 units and a height of 1000 units in the network simulation.

set nf [open lab2.nam w]

$ns namtrace-all-wireless **$nf** 1000 1000

* In an ns-2 script involve the **creation of a nam (Network Animator) trace file and configuring wireless tracing for visualization.**
* set nf [open lab2.nam w]: This line opens a file named lab2.nam in write mode (w) and **assigns the file handle to the variable nf.** This file will be used to **store trace information for visualization using nam**.
* $ns namtrace-all-wireless $nf 1000 1000: This line **configures nam to trace all wireless events** and writes the trace information to the file opened earlier (lab2.nam). The 1000 1000 arguments specify **the dimensions of the visualization window in nam.**
* $ns namtrace-all-wireless: **Enables wireless tracing for all nodes and their wireless communication.**
* $nf: The file handle for the nam trace file.
* 1000 1000: Specifies the dimensions of the visualization window in nam. In this case, the window is set to 1000x1000 units.

**#specify node configuration**

* Configure the characteristics of the nodes in your wireless ad-hoc network simulation.

-adhocRouting DSDV: Sets the ad-hoc routing protocol to **DSDV (Destination Sequenced Distance Vector)**. DSDV is a proactive routing protocol used in wireless ad-hoc networks.

-llType LL: Specifies the **link layer type as LL (Link Layer).**

-macType Mac/802\_11: Specifies the Medium Access Control (MAC) layer type as Mac/802\_11, indicating the use of the **IEEE 802.11 MAC protocol** commonly used in wireless networks.

-ifqType Queue/DropTail: **Sets the interface queue type** to Queue/DropTail, which is a simple drop-tail queue.

-ifqLen 20: **Sets the interface queue length** to 20.

-phyType Phy/WirelessPhy: **Specifies the physical layer type as Phy/WirelessPhy.**

-channelType Channel/WirelessChannel: **Sets the channel type** to Channel/WirelessChannel.

-propType Propagation/TwoRayGround: **Specifies the propagation model** as Propagation/TwoRayGround.

-antType Antenna/OmniAntenna: **Sets the antenna type** to Antenna/OmniAntenna, indicating an omnidirectional antenna.

-topoInstance $topo: **Associates** the topology instance ($topo) with the node configuration.

-agentTrace ON: Enables tracing for agents.

-routerTrace ON: Enables tracing for routers.

**create god-4**

* To create a god object in ns-2. A god object is a **special agent that has visibility into the entire network** and can **control or monitor the activities of all other nodes** in the simulation.
* The line create-god 4 suggests that we are **creating a god object for a network with four nodes.** This god object is used for various purposes, including routing and monitoring.

$n0 set X\_ 110

$n0 set Y\_ 500

$n0 set Z\_ 0

* We are **configuring the spatial coordinates of the node $n0** in your ns-2 wireless ad-hoc network simulation.
* $n0 set X\_ 110: **Sets the X-coordinate** of node $n0 to 110. This likely represents the horizontal position of the node in the simulation.
* $n0 set Y\_ 500: **Sets the Y-coordinate** of node $n0 to 500. This likely represents the vertical position of the node in the simulation.
* $n0 set Z\_ 0: **Sets the Z-coordinate** of node $n0 to 0. The Z-coordinate is often used for the **elevation of the node in a three-dimensional space**. In this case, the value 0 suggests that the node is at ground level.

**set tcp0 [new Agent/TCP]**

**$ns attach-agent $n0 $tcp0**

**set ftp0 [new Application/FTP]**

**$ftp0 attach-agent $tcp0**

**set sink1 [new Agent/TCPSink]**

**$ns attach-agent $n1 $sink1**

**$ns connect $tcp0 $sink1**

The snippet **sets up a simple TCP connection between two nodes ($n0 and $n1) using agents and applications.**

* set tcp0 [new Agent/TCP]: **Creates a new TCP agent** named $tcp0.
* $ns attach-agent $n0 $tcp0: **Attaches the TCP agent** $tcp0 to node $n0. This associates the TCP agent with the first node in the simulation.
* set ftp0 [new Application/FTP]: **Creates a new FTP application** named $ftp0.
* $ftp0 attach-agent $tcp0: **Attaches the FTP application to the TCP agent** $tcp0. This means that the FTP application will use the TCP agent for communication.
* set sink1 [new Agent/TCPSink]: **Creates a TCP sink agent** named $sink1. A TCP sink is used to receive TCP data.
* $ns attach-agent $n1 $sink1: **Attaches the TCP sink** agent $sink1 to node $n1. This associates the TCP sink with the second node in the simulation.
* $ns connect $tcp0 $sink1: **Connects the TCP agent $tcp0 (sender) to the TCP sink agent $sink1 (receiver).** This establishes a TCP connection between the two nodes.
* In summary, this script **sets up a basic TCP connection between nodes $n0 and $n1 using a TCP agent, an FTP application, and a TCP sink.**
* The FTP application **generates traffic that is sent by the TCP agent** on node $n0 and received by the TCP sink on node $n1. This is a simple example of a **client-server communication setup using TCP in ns-2.**

**$ns at 0.5 "$ftp0 start"**

The line $ns at 0.5 "$ftp0 start" **schedules the start of the FTP application** ($ftp0) at simulation time 0.5 seconds.

**#set up a destination for mobile nodes. They move to <x><y> coordinates at <s>m/s.**

* These lines **schedule movements for nodes** in your wireless ad-hoc network simulation using the **setdest mobility model.**
* At time 0.3 seconds, **nodes $n0, $n1, $n2, and $n3 are assigned initial destinations** using the setdest command. The format is "$node setdest X Y Speed", **where X and Y are the coordinates, and Speed is the speed of the node.**
* At time 10.0 seconds, **nodes $n0 and $n1 are assigned new destinations.**
* At time 70.0 seconds, **nodes $n0 and $n1 are assigned new destinations.**
* At various later times, **additional movements are scheduled for nodes $n0 and $n1.**

**Explanation for AWK file**

* BEGIN block: **Initializes variables** before processing the file.
* Main block: **Processes each line of the input file**. The conditions inside the block identify lines corresponding to the **sending of TCP packets** (s, AGT, tcp), **receiving of TCP packets** (r, AGT, tcp), and **packets arriving at the router** (s, RTR, tcp). **Counters** (PktsSent, PktsRcvd, PktsAtRTR) are updated accordingly.
* END block: **Prints the calculated metrics** based on the data processed from the trace file. Metrics include the **number of packets sent, the number of packets received, packet delivery ratio, and routing load.**