Program 2:

Simulate simple ESS and with transmitting nodes in wireless LAN by simulation and determine the performance with respect to transmission of packets.

```
set ns [new Simulator]
set tf [open lab8.tr w]
$ns trace-all $tf
set topo [new Topography]
$topoload_flatgrid 1000 1000
set nf [open lab8.nam w]
$ns namtrace-all-wireless $nf 1000 1000
$ns node-config -adhocRouting DSDV \
-llType LL \
-macType Mac/802_11 \
-ifqType Queue/DropTail \
-ifqLen 50 \
-phyTypePhy/WirelessPhy \
-channelType Channel/WirelessChannel \
-prrootype Propagation/TwoRayGround \
-antType Antenna/OmniAntenna \
-topoInstance $topo \
-agentTrace ON \
-routerTrace ON
create-god 3
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
$n0 label "tcp0"
$n1 label "sink1/tcp1"
$n2 label "sink2"
$n0 set X_ 50
$n0 set Y_ 50
$n0 set Z_ 0
$n1 set X_ 100
$n1 set Y_ 100
$n1 set Z_ 0
$n2 set X_ 600
$n2 set Y_ 600
$n2 set Z_ 0
$ns at 0.1 "$n0 setdest 50 50 15"
$ns at 0.1 "$n1 setdest 100 100 25"
$ns at 0.1 "$n2 setdest 600 600 25"
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
set tcp1 [new Agent/TCP]
$ns attach-agent $n1 $tcp1
set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1
set sink2 [new Agent/TCPSink]
$ns attach-agent $n2 $sink2
$ns connect $tcp1 $sink2
$ns at 5 "$ftp0 start"
$ns at 5 "$ftp1 start"
$ns at 100 "$n1 setdest 550 550 15"
$ns at 190 "$n1 setdest 70 70 15"
proc finish { } {
global ns nftf
$ns flush-trace
exec nam lab8.nam &
close $tf
exit 0
}
$ns at 250 "finish"
$ns run
```

This code is a script written in the Tcl scripting language for the Network Simulator (NS-2). NS-2 is a discrete event simulator used for simulating network protocols and scenarios.

Sl.no	Command	Function
1	set ns [new Simulator]	Creates a new simulation object (ns) of the Simulator class.
2	set tf [open lab8.tr w]	Opens a new file named "lab8.tr" for writing. This file will be used to store the simulation trace.
3	`$ns trace-all $tf	Configures the simulator to trace all events and write the trace to the "lab8.tr" file.
4	set topo [new Topography]	Creates a new topography object (topo) to represent the simulated environment.
5	$topoload_flatgrid 1000 1000	Loads a flat grid of size 1000x1000 in the topography
6	set nf [open lab8.nam w]	Opens a new file named "lab8.nam" for writing. This file will be used for network animation (using the Network Animator - Nam).
7	$ns namtrace-all-wireless $nf 1000 1000	Configures the simulator to trace all wireless events and write the trace to the

		"lab8.nam" file. The simulation area is set to 1000x1000.
8	$ns node-config -adhocRouting DSDV \\ -llType LL \\ -macType Mac/802_11 \\ -ifqType Queue/DropTail \\ -ifqLen 50 \\ -phyTypePhy/WirelessPhy \\ -channelType Channel/WirelessChannel \\ -prrootype Propagation/TwoRayGround \\ -antType Antenna/OmniAntenna \\ -topoInstance $topo \\ -agentTrace ON \\ -routerTrace ON	Configures the nodes in the simulation with various parameters, such as ad-hoc routing protocol (DSDV), link layer type, MAC type, queue type, physical layer type, etc.
9	create-god 3	Creates a God object to manage global information in the simulation for 3 nodes
10	set n0 [$ns node] set n1 [$ns node] set n2 [$ns node]	Creates three nodes (n0, n1, and n2) in the simulation.
11	$n0 label "tcp0" $n1 label "sink1/tcp1" $n2 label "sink2"	Labels the nodes for better identification in the simulation.
12	$n0 set X_ 50 $n0 set Y_ 50 $n0 set Z_ 0 $n1 set X_ 100 $n1 set Y_ 100 $n1 set Z_ 0 $n2 set X_ 600 $n2 set Y_ 600 $n2 set Z_ 0	Setting initial positions for the nodes
13	$ns at 0.1 "$n0 setdest 50 50 15" $ns at 0.1 "$n1 setdest 100 100 25" $ns at 0.1 "$n2 setdest 600 600 25"	Sets the initial destination for node n0 to (50, 50, 15) at simulation time 0.1.
14	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 set tcp1 [new Agent/TCP] $ns attach-agent $n1 $tcp1 set ftp1 [new Application/FTP] $ftp1 attach-agent $tcp1 set sink2 [new Agent/TCPSink] $ns attach-agent $n2 $sink2 $ns connect $tcp1 $sink2	Agents and applications are configured for communication between nodes using TCP and FTP **Detailed Explanation at Table 2**

	$ns at 5 "$ftp0 start" $ns at 5 "$ftp1 start"	
15	$ns at 100 "$n1 setdest 550 550 15" $ns at 190 "$n1 setdest 70 70 15"	behavior of node n1 during the course of the simulation. The destinations specified in these lines represent the new positions to which node n1will move at the specified simulation times. 1. $ns at 100 "$n1 setdest 550 550 15": At simulation time 100, it sets the destination of node n1to (550, 550, 15). This means that at time 100, node n1 will change its destination to (550, 550, 15). 2. $ns at 190 "$n1 setdest 70 70 15": At simulation time 190, it sets the destination of node n1 to (70, 70, 15). This means that at time 190, node n1 will change its destination to (70, 70, 15).

16	proc finish { } {
	global ns nftf
	$ns flush-trace
	exec nam lab8.nam &
	close $tf
	exit 0
	}
		Defines a procedure called finis` to perform cleanup tasks when the simulation ends.

17	$ns at 250 "finish"	Schedules thefinishprocedure to be executed at simulation time 250
18	$ns run	Initiates the simulation.

The code sets up a wireless ad-hoc network with three nodes, configures their initial positions, defines communication between nodes using TCP and FTP, and schedules the simulation to run until time 250, at which point the `finish` procedure is called to perform cleanup tasks. The simulation trace is written to "lab8.tr," and the network animation trace is written to "lab8.nam."

Table 2:

Sl. no	Lines of code	Detailed Explaination
1	-llType LL \	-Specifies the link layer type. In

	-macType Mac/802_11 \ -ifqType Queue/DropTail \ -ifqLen 50 \ -phyTypePhy/WirelessPhy \ -channelType Channel/WirelessChannel \ -prrootype Propagation/TwoRayGround \ -antType Antenna/OmniAntenna \ -topoInstance $topo \ -agentTrace ON \ -routerTrace ON	this case, it's set to "LL" (Link Layer). - Specifies the MAC (Medium Access Control) layer type. It's set to "Mac/802_11," indicating the use of the IEEE 802.11 standard for wireless LANs. - Specifies the type of interface queue. In this case, it's set to a drop-tail queue. - Sets the length of the interface queue to 50. This parameter defines the maximum number of packets that can be enqueued at the interface. - Specifies the physical layer type. It's set to "Phy/WirelessPhy," indicating the use of a wireless physical layer. - Specifies the type of channel. It's set to "Channel/WirelessChannel," indicating the use of a wireless channel. - Specifies the type of propagation model. In this case, it's set to "Propagation/TwoRayGround," which is a commonly used propagation model for wireless communication. - Specifies the type of antenna. It's set to "Antenna/OmniAntenna," indicating the use of an omnidirectional antenna. - Specifies the topology instance to be used. It uses the previously created topography object `$topo` for the simulation. - Enables tracing for agents. This means that information about agent activities will be recorded in the simulation trace - Enables tracing for routers. This means that information

		about router activities will be recorded in the simulation trace.
2	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 set tcp1 [new Agent/TCP] $ns attach-agent $n1 $tcp1 set ftp1 [new Application/FTP] $ftp1 attach-agent $tcp1 set sink2 [new Agent/TCPSink] $ns attach-agent $n2 $sink2 $ns connect $tcp1 $sink2 $ns at 5 "$ftp0 start" $ns at 5 "$ftp1 start"	**set up TCP agents, FTP applications, and TCP sinks for communication between nodes** 1. set tcp0 [new Agent/TCP]: Creates a new TCP agent object namedtcp0 2. Attaches the TCP agent tcp0 to node n0. This means that node n0 will use this TCP agent for communication. 3. Creates a new FTP application object named ftp0. 4. This means that the FTP application will use the TCP agent for sending data. 5. Creates a new TCP sink agent object named sink1. A TCP sink is used to receive data. 6. This means that node n1 will use this TCP sink for receiving data. 7. This establishes the communication link between node n0 (sending data) and node n1 (receiving data). 8. Creates a new TCP agent object named tcp1 Attaches the TCP agent tcp1 to node n1. Creates a new FTP application object named ftp1. Attaches the TCP agent tcp1 to the FTP application ftp1. Creates a new TCP sink agent object named sink2. Attaches the TCP sink agent

		sink2 to node n2. Connects the TCP agent tcp1to the TCP sink sink2. 9. Schedules the FTP application ftp0 to start sending data at simulation time 5. 10. Schedules the FTP application ftp1 to start sending data at simulation time 5. These lines essentially set up two TCP connections with FTP applications, where tcp0 and ftp0 are associated with the communication between nodes n0 and n1, and tcp1 and ftp1 are associated with the communication between nodes n1 and n2. The data transmission is scheduled to start at simulation time 5.

AWK file: *(Open a new editor using "vi command" and write awk file and save with ".awk" extension)*

```
BEGIN{
count1=0
count2=0
pack1=0
pack2=0
time1=0
time2=0
}
{ if($1= ="r"&& $3= ="_1_" && $4= ="AGT")
{ count1++
pack1=pack1+$8
time1=$2 }
if($1= ="r" && $3= ="_2_" && $4= ="AGT")
{ count2++

pack2=pack2+$8
time2=$2 }
}
END{
printf("The Throughput from n0 to n1: %f Mbps \n",
((count1*pack1*8)/(time1*1000000)));
printf("The Throughput from n1 to n2: %f Mbps", ((count2*pack2*8)/(time2*1000000)));
}
```

Output:

Steps for execution

- *Open vi editor and type program. Program name should have the extension " **.tcl** "*

  [root@localhost ~]# vi lab8.tcl

- *Save the program by pressing **"ESC key"** first, followed by **"Shift and :"** keys simultaneously and type **"wq"** and press **Enter key**.*
- *Open vi editor and type **awk** program. Program name should have the extension "**.awk** "*

  [root@localhost ~]# vi lab8.awk

- *Save the program by pressing **"ESC key"** first, followed by **"Shift and :"** keys simultaneously and type **"wq"** and press **Enter key**.*
- *Run the simulation program*

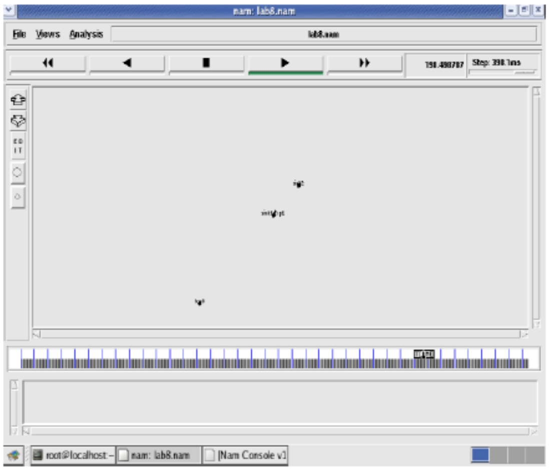
  [root@localhost~]# ns lab8.tcl

  - *Here **"ns"** indicates network simulator. We get the topology shown in the snapshot.*
  - *Now press the play button in the simulation window and the simulation will begins.*
- *After simulation is completed run **awk file** to see the output ,*

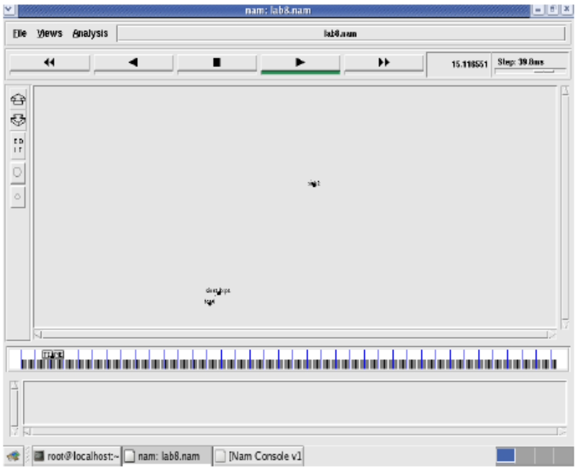
  [root@localhost~]# awk –f lab8.awk lab8.tr

- *To see the trace file contents open the file as ,*

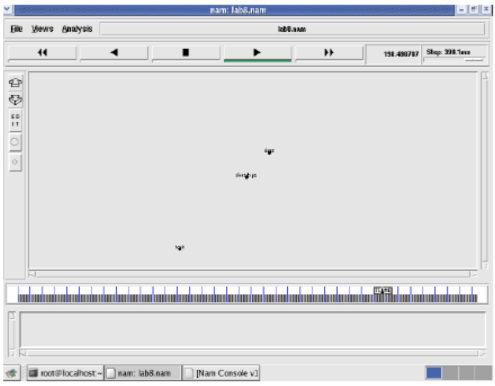
  [root@localhost~]# vi lab8.tr

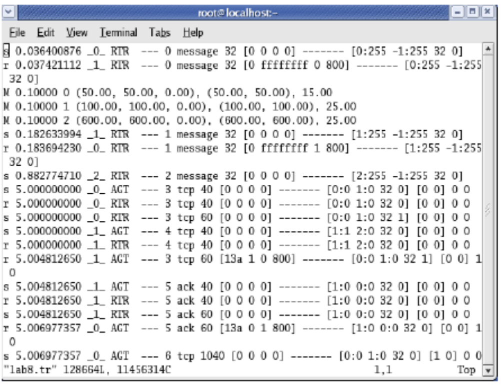


Node 1 and 2 are communicating



Node 2 is moving towards node 3

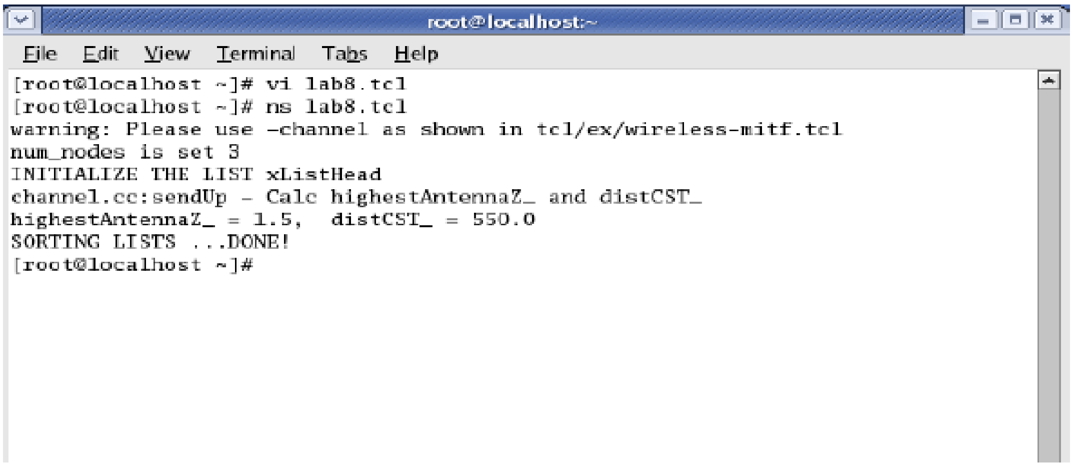




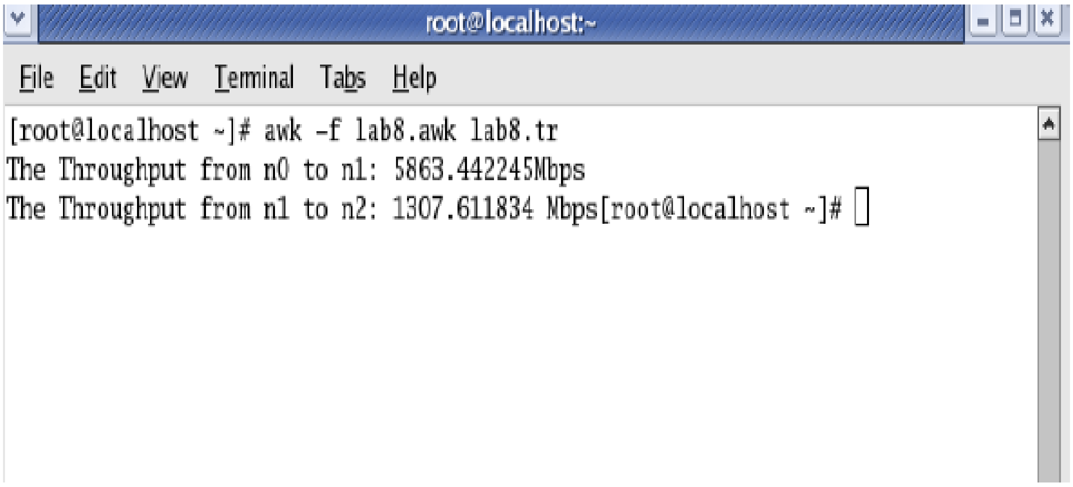
```
s 0.036400876 _0_ RTR  --- 0 message 32 [0 0 0 0] ------- [0:255 -1:255 32 0]
r 0.037421112 _1_ RTR  --- 0 message 32 [0 ffffffff 0 800] ------- [0:255 -1:255
 32 0]
M 0.10000 0 (50.00, 50.00, 0.00), (50.00, 50.00), 15.00
M 0.10000 1 (100.00, 100.00, 0.00), (100.00, 100.00), 25.00
M 0.10000 2 (600.00, 600.00, 0.00), (600.00, 600.00), 25.00
s 0.182633994 _1_ RTR  --- 1 message 32 [0 0 0 0] ------- [1:255 -1:255 32 0]
r 0.183694230 _0_ RTR  --- 1 message 32 [0 ffffffff 1 800] ------- [1:255 -1:255
 32 0]
s 0.882774710 _2_ RTR  --- 2 message 32 [0 0 0 0] ------- [2:255 -1:255 32 0]
s 5.000000000 _0_ AGT  --- 3 tcp 40 [0 0 0 0] ------- [0:0 1:0 32 0] [0 0] 0 0
r 5.000000000 _0_ RTR  --- 3 tcp 40 [0 0 0 0] ------- [0:0 1:0 32 0] [0 0] 0 0
s 5.000000000 _0_ RTR  --- 3 tcp 60 [0 0 0 0] ------- [0:0 1:0 32 1] [0 0] 0 0
s 5.000000000 _1_ AGT  --- 4 tcp 40 [0 0 0 0] ------- [1:1 2:0 32 0] [0 0] 0 0
r 5.000000000 _1_ RTR  --- 4 tcp 40 [0 0 0 0] ------- [1:1 2:0 32 0] [0 0] 0 0
r 5.004812650 _1_ AGT  --- 3 tcp 60 [13a 1 0 800] ------- [0:0 1:0 32 1] [0 0] 1
 0
s 5.004812650 _1_ AGT  --- 5 ack 40 [0 0 0 0] ------- [1:0 0:0 32 0] [0 0] 0 0
r 5.004812650 _1_ RTR  --- 5 ack 40 [0 0 0 0] ------- [1:0 0:0 32 0] [0 0] 0 0
s 5.004812650 _1_ RTR  --- 5 ack 60 [0 0 0 0] ------- [1:0 0:0 32 0] [0 0] 0 0
r 5.006977357 _0_ AGT  --- 5 ack 60 [13a 0 1 800] ------- [1:0 0:0 32 0] [0 0] 1
 0
s 5.006977357 _0_ AGT  --- 6 tcp 1040 [0 0 0 0] ------- [0:0 1:0 32 0] [1 0] 0 0
"lab8.tr" 128664L, 11456314C                                     1,1           Top
```

Node 2 is coming back from node 3 towards node1 Trace File

Here **"M"** indicates mobile nodes, **"AGT"** indicates Agent Trace, **"RTR"** indicates Router Trace



```
[root@localhost ~]# vi lab8.tcl
[root@localhost ~]# ns lab8.tcl
warning: Please use -channel as shown in tcl/ex/wireless-mitf.tcl
num_nodes is set 3
INITIALIZE THE LIST xListHead
channel.cc:sendUp - Calc highestAntennaZ_ and distCST_
highestAntennaZ_ = 1.5,  distCST_ = 550.0
SORTING LISTS ...DONE!
[root@localhost ~]#
```



```
[root@localhost ~]# awk -f lab8.awk lab8.tr
The Throughput from n0 to n1: 5863.442245Mbps
The Throughput from n1 to n2: 1307.611834 Mbps[root@localhost ~]#
```