# Experiment No. 7

**<u>Aim</u>:** To analyze the performance of a network for QoS (quality of service) parameters.

## <u>Implementation:</u>

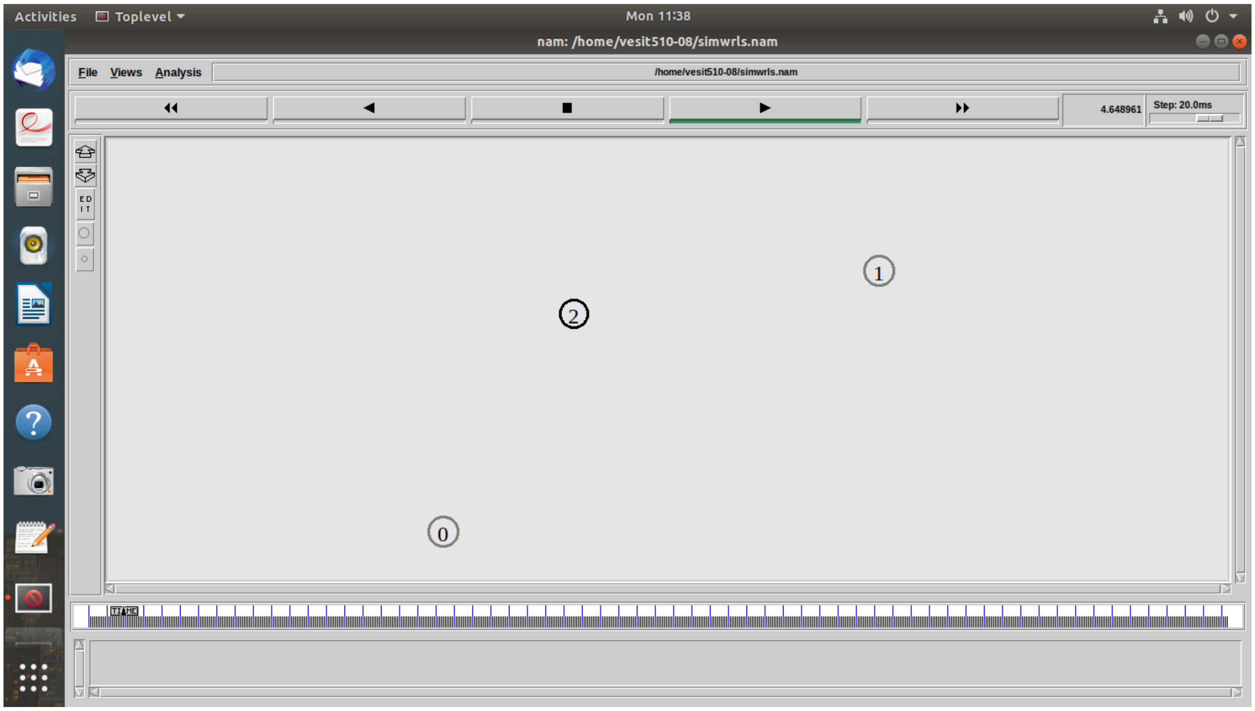
1. **Wireless Propagation**

**Code:**

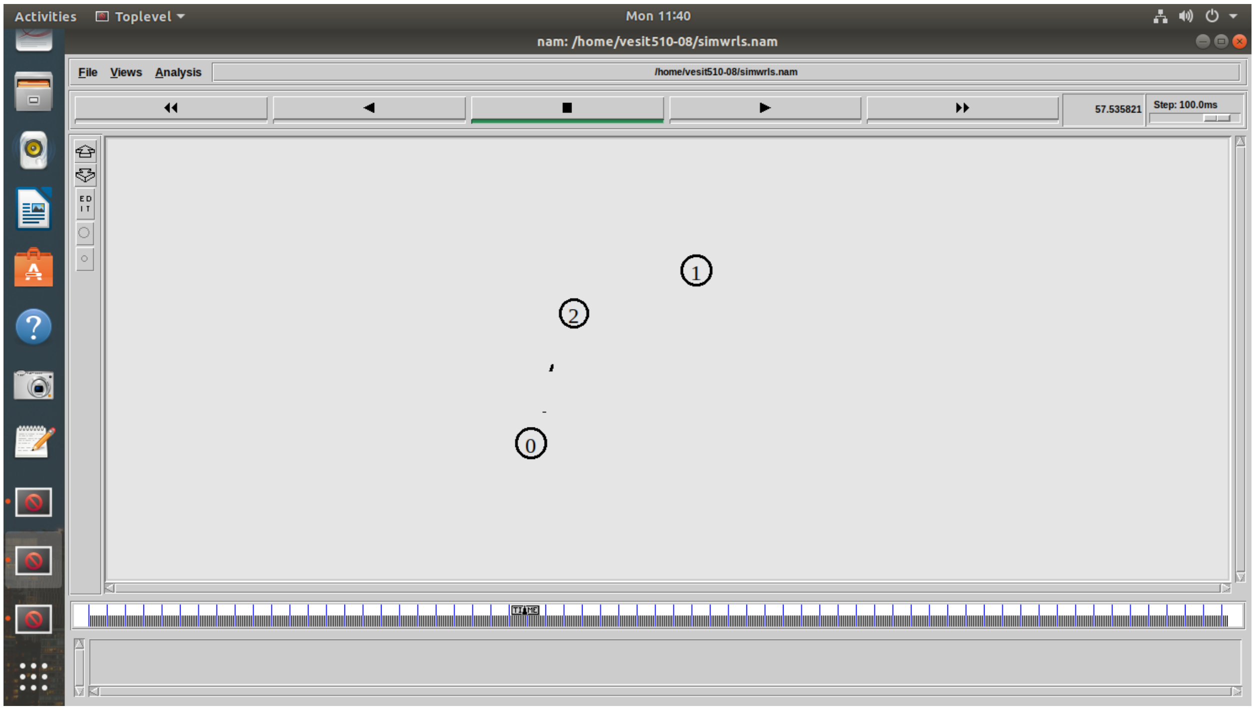
```
#Wireless Network
#Setting values for variables in an associative array: val
#chan: Channel Type: Wireless
#prop: Radio Propagation Model: Two way propagation
#netif: Network Interface Types: Wireless
#mac: MAC type: Cellular Communication
#ifq: Interface Queue type
#ll: Link Layer type
#ant: Antenna Type
#ifqlen: Interface queue length
#nn: Number of nodes
#rp: ad-hoc routing protocol: Destination Sequenced Distance Vector (DSDV) #X,
Y: Positions
#Stop: Stop time
set val(chan) Channel/WirelessChannel
set val(prop) Propagation/TwoRayGround ;
set val(netif) Phy/WirelessPhy ;
set val(mac) Mac/802_11 ;
set val(ifq) Queue/DropTail/PriQueue ;
set val(ll) LL ;
set val(ant) Antenna/OmniAntenna ;
set val(ifqlen) 50 ;
set val(nn) 3 ;
set val(rp) DSDV ;
set val(x) 500 ;
set val(y) 400 ;
set val(stop) 150 ;
#Create simulator object and link the trace files and nam trace
set ns [new Simulator]
set tracefd [open simple-dsdv.tr w]
set namtrace [open simwrls.nam w]
#Linking trace files to trace buffers
$ns trace-all $tracefd
$ns use-newtrace
$ns namtrace-all-wireless $namtrace $val(x) $val(y)
#Create topography flatgrid refers to movement in XY plane
set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)
#Create General Operations Director(GOD) object
#GOD object stores total number of mobile nodes & table of shortest hops required
create-god $val(nn)
#Configuring nodes
# agentTrace: tracing at agent level turned ON or OFF
# routerTrace: tracing at router level turned ON or OFF
# macTrace: tracing at mac level turned ON or OFF
$ns node-config -adhocRouting $val(rp) \
-llType $val(ll) \
-macType $val(mac) \
-ifqType $val(ifq) \
-ifqLen $val(ifqlen) \
-antType $val(ant) \
-propType $val(prop) \
-phyType $val(netif) \
-channelType $val(chan) \
-topoInstance $topo \
-agentTrace ON \
-routerTrace ON \
-macTrace OFF \
-movementTrace ON
#Creating three nodes
#Setting node positions, z=0 as topology is flatgrid
```

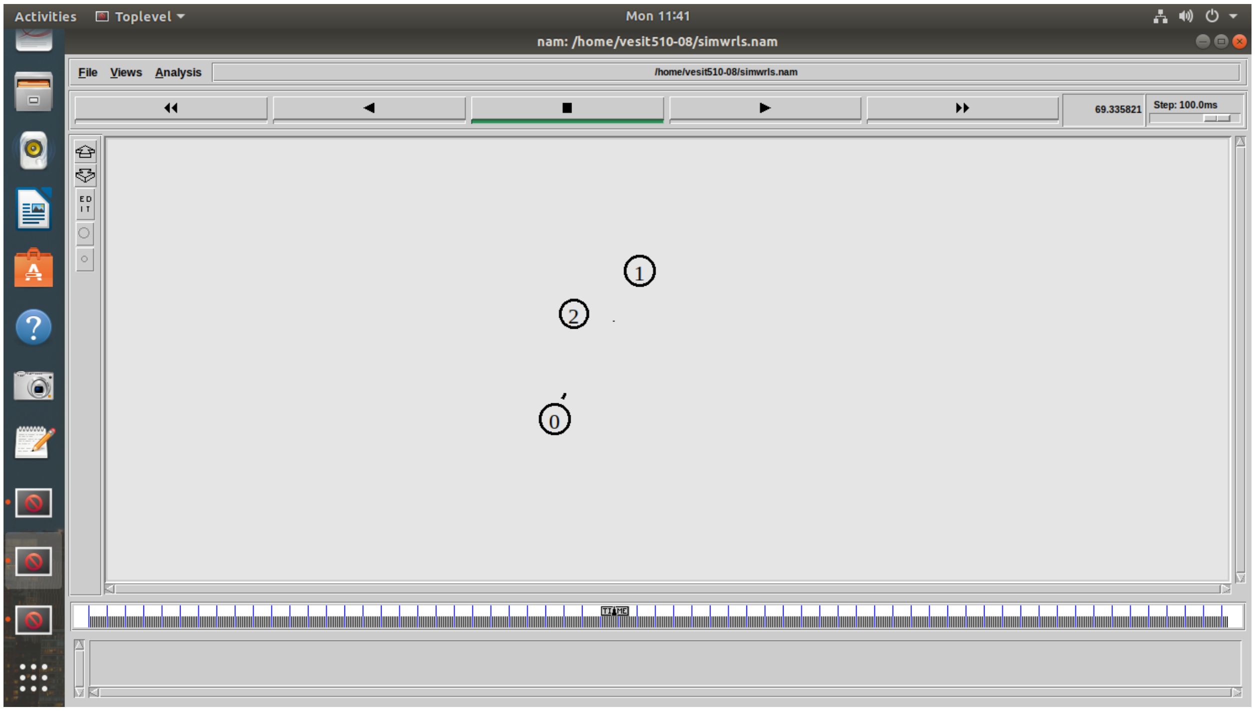
```
for {set i 0} {$i < $val(nn) } { incr i } {
set node_($i) [$ns node]
}
#Setting node mobility
$node_(0) set X_ 5.0
$node_(0) set Y_ 5.0
$node_(0) set Z_ 0.0
$node_(1) set X_ 490.0
$node_(1) set Y_ 285.0
$node_(1) set Z_ 0.0
$node_(2) set X_ 150.0
$node_(2) set Y_ 240.0
$node_(2) set Z_ 0.0
# setdest params: X, Y, speed.
$ns at 10.0 "$node_(0) setdest 250.0 250.0 3.0"
$ns at 15.0 "$node_(1) setdest 45.0 285.0 5.0"
$ns at 110.0 "$node_(0) setdest 480.0 300.0 5.0"
#Attaching transport layer protocol agents and application layer protocol agents to the nodes
#Setting source node, sink node
set tcp [new Agent/TCP/Newreno]
set sink [new Agent/TCPSink]
$ns attach-agent $node_(0) $tcp
$ns attach-agent $node_(1) $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 10.0 "$ftp start"
#defines Node initial position. 30 is node size Must be called after mobility
for {set i 0} {$i < $val(nn)} { incr i } {
   $ns initial_node_pos $node_($i) 30
}
#Reset positions at stop
for {set i 0} {$i < $val(nn) } { incr i } {
   $ns at $val(stop) "$node_($i) reset";
}
#Start and stop wireless simulation
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "stop"
$ns at 150.01 "puts \"end simulation\" ; $ns halt"
proc stop {} {
#Flush trace buffers, close files and execute nam file
   global ns tracefd namtrace
   $ns flush-trace
   close $tracefd
   close $namtrace
   exec nam simwrls.nam &
   exit
}
$ns run
```

**Output:**

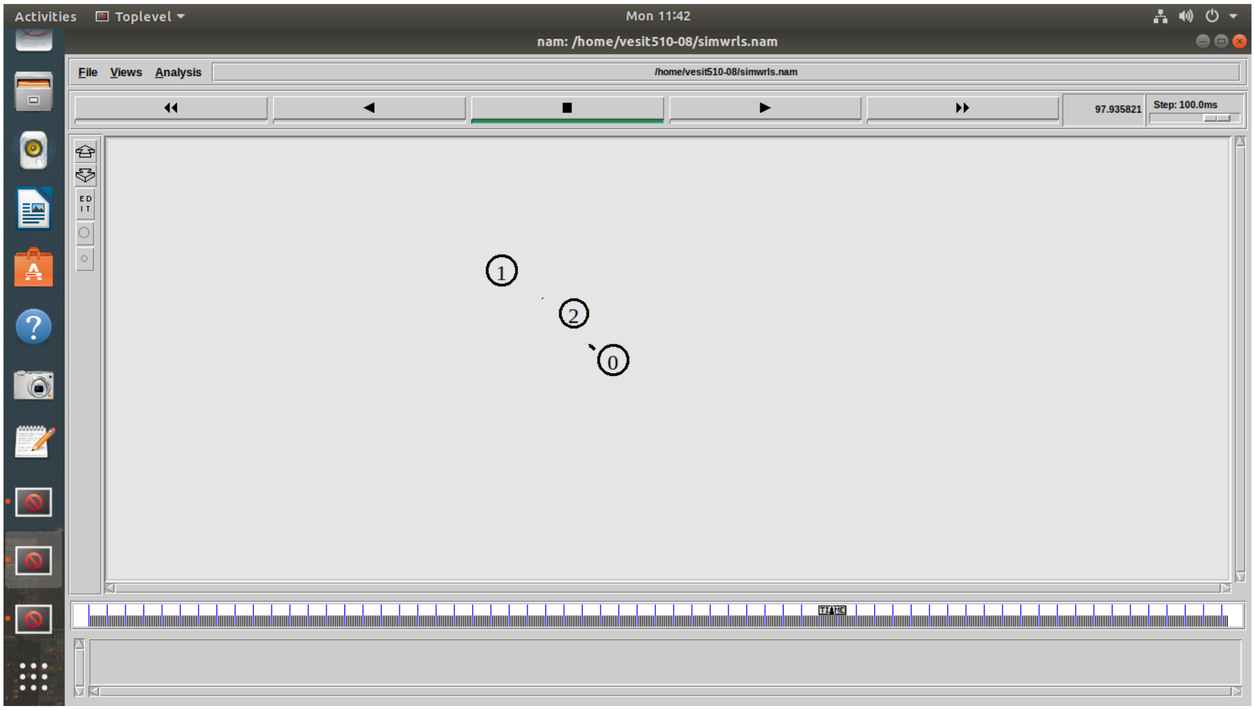


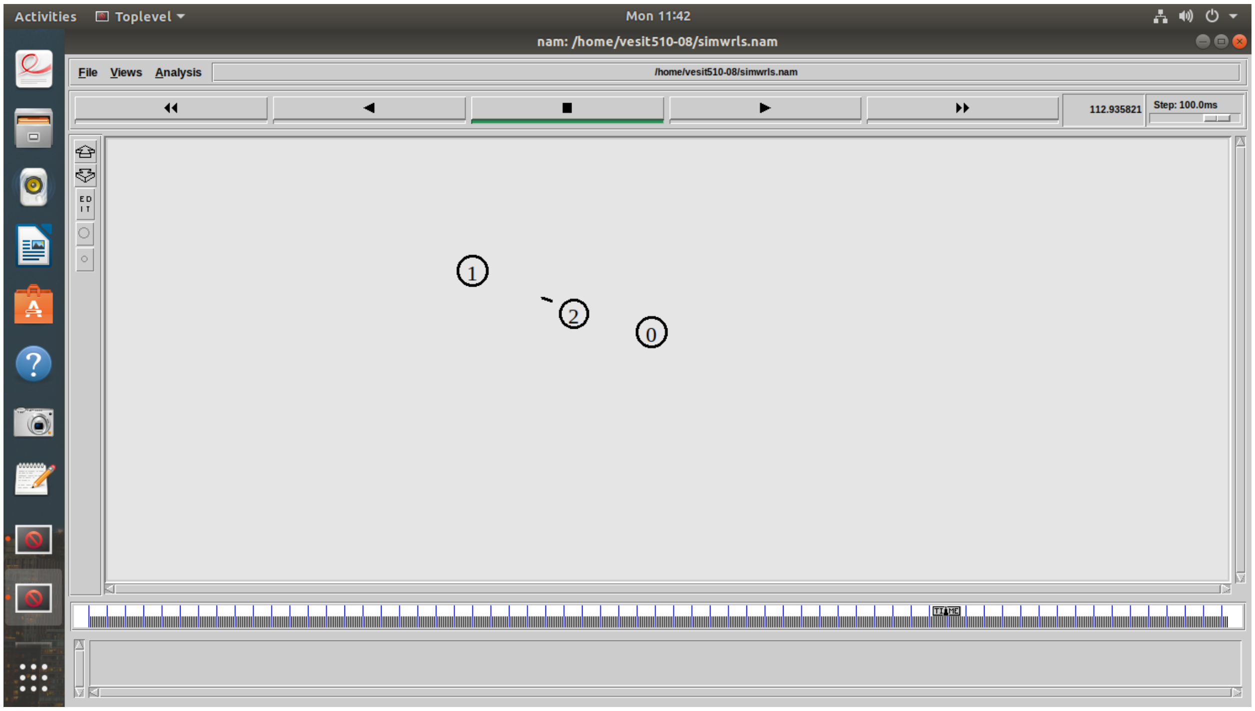
Manav_17


Activities
Toplevel
Mon 11:40
nam: /home/vesit510-08/simwrls.nam
File
Views
Analysis
/home/vesit510-08/simwrls.nam
57.535821
Step: 100.0ms
E D I T
1
2
0

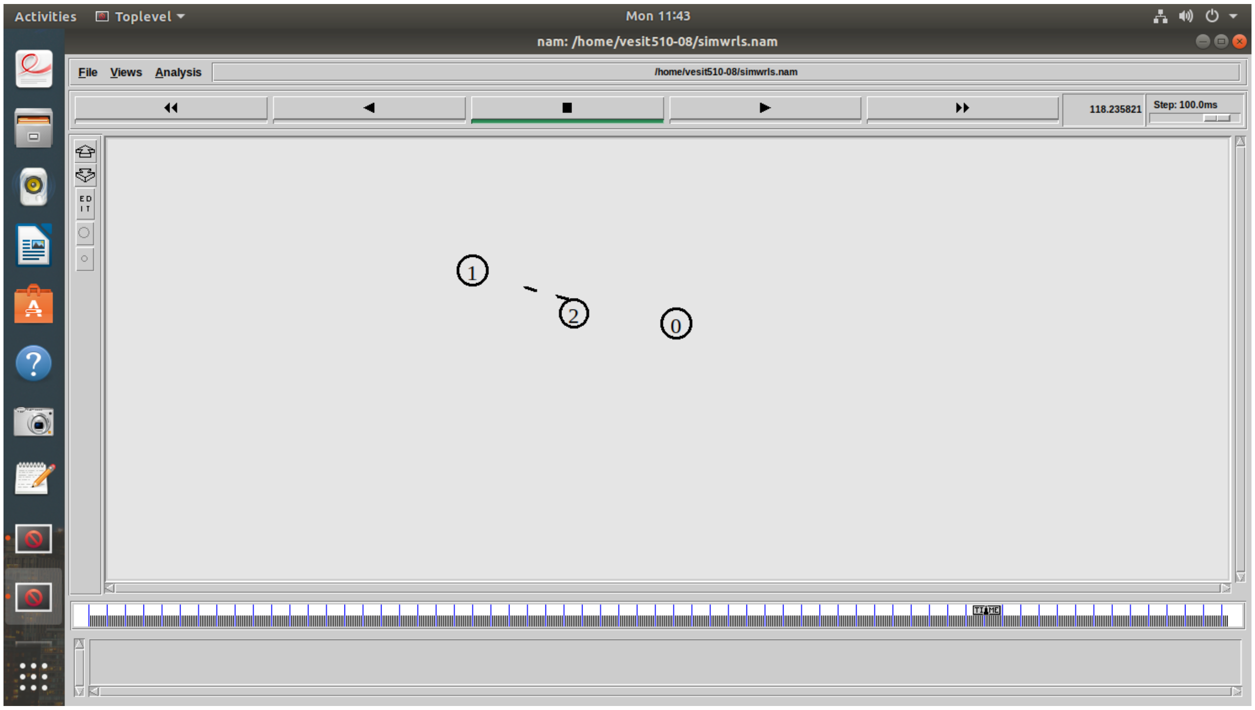

Activities
Toplevel
Mon 11:41
nam: /home/vesit510-08/simwrls.nam
File
Views
Analysis
/home/vesit510-08/simwrls.nam
69.335821
Step: 100.0ms
E D I T
1
2
0

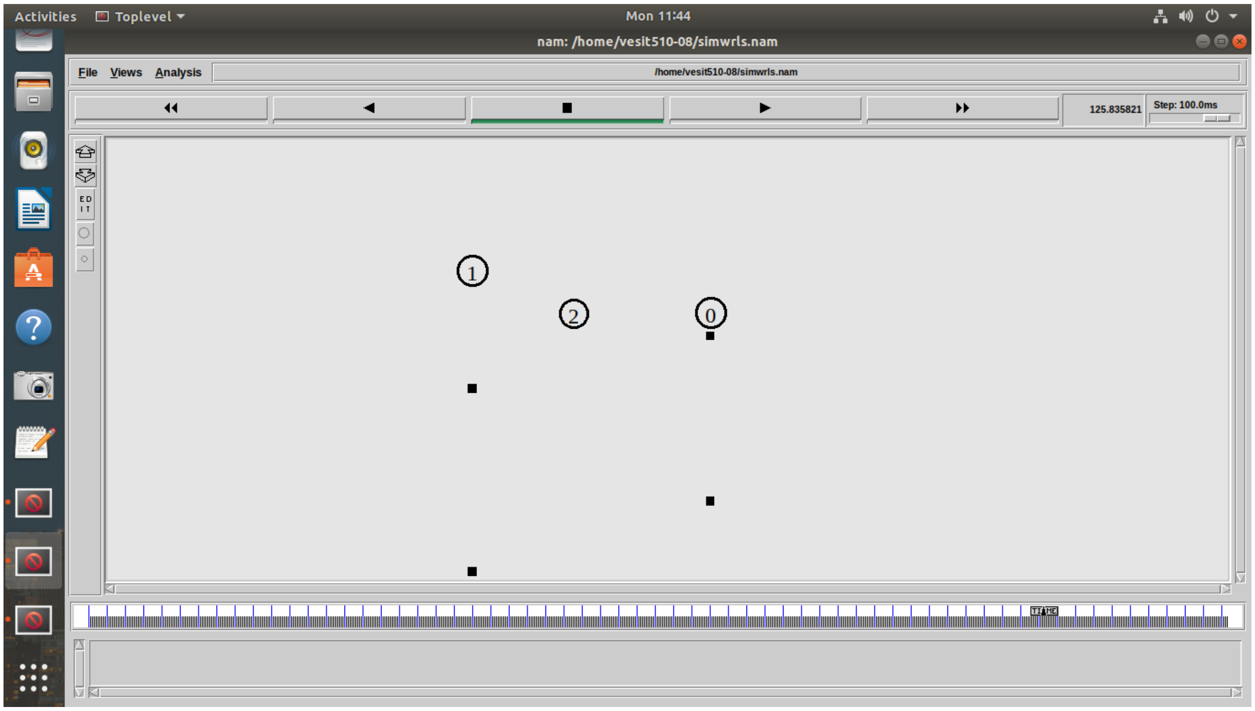
Manav_17

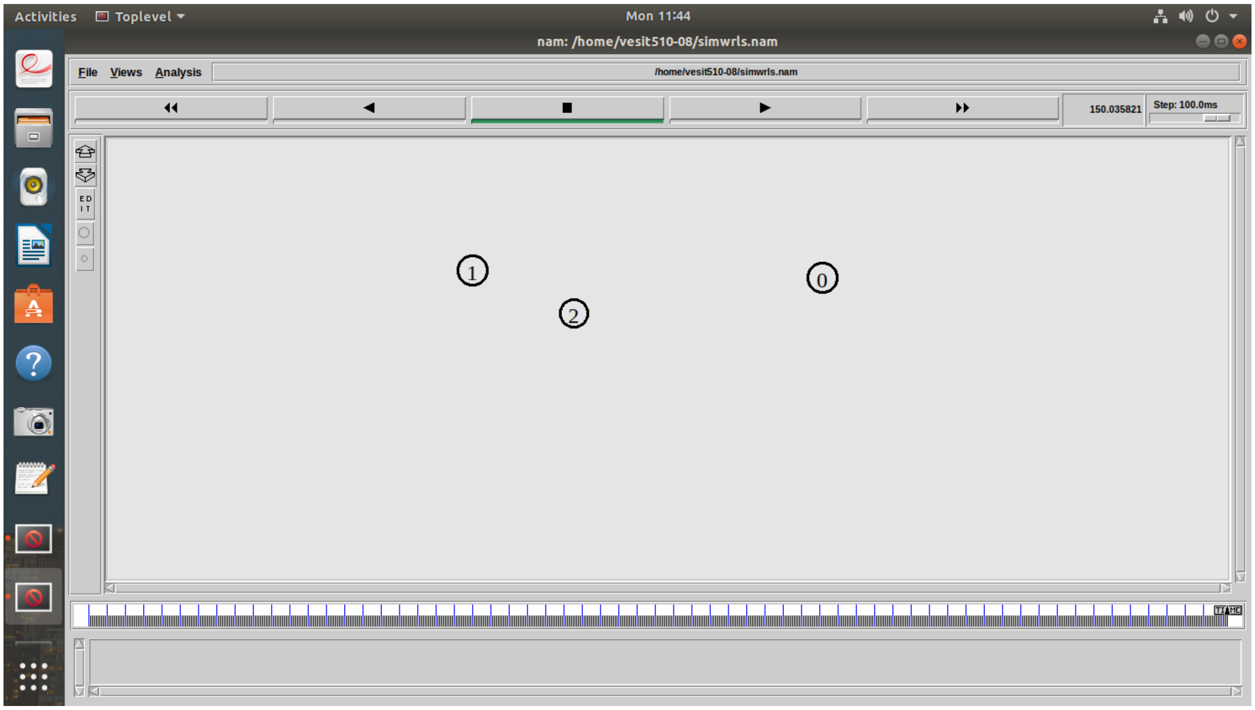

Activities
Toplevel
Mon 11:42
nam: /home/vesit510-08/simwrls.nam
File
Views
Analysis
/home/vesit510-08/simwrls.nam
97.935821
Step: 100.0ms
EDIT
1
2
0


Activities
Toplevel
Mon 11:42
nam: /home/vesit510-08/simwrls.nam
File
Views
Analysis
/home/vesit510-08/simwrls.nam
112.935821
Step: 100.0ms
EDIT
1
2
0

Manav_17


Activities
Toplevel
Mon 11:43
nam: /home/vesit510-08/simwrls.nam
File
Views
Analysis
/home/vesit510-08/simwrls.nam
118.235821
Step: 100.0ms
EDIT
1
2
0


Activities
Toplevel
Mon 11:44
nam: /home/vesit510-08/simwrls.nam
File
Views
Analysis
/home/vesit510-08/simwrls.nam
125.835821
Step: 100.0ms
EDIT
1
2
0



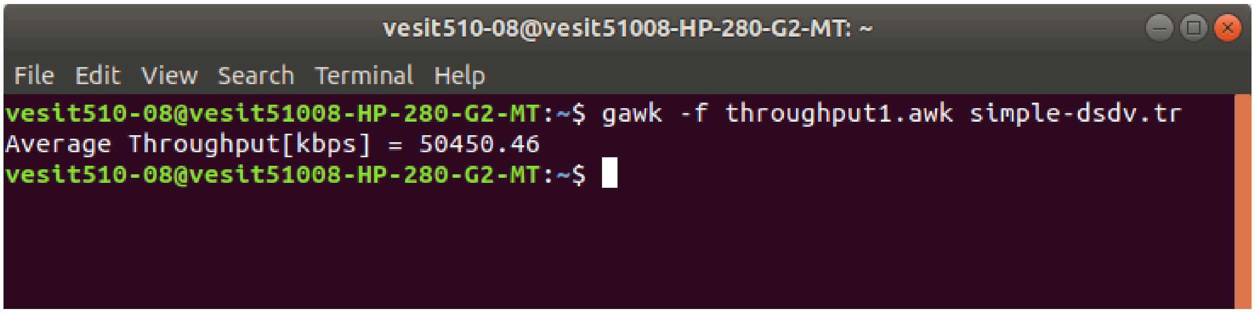
## 2. Getting Throughput

**Code:**

```
BEGIN {
recvdSize = 0 # received packet size
startTime = 400 # high random start time
stopTime = 5 #low random stop time
}
{
#Analyze the trace file
event = $1 # send or received (s/r)
time = $3 # time of transaction (time of sending)
pkt_size = $37 # packet size
level = $19 # application agent or routing protocol data (AGT/RTR)
# Find the starting time of simulation

if (level == "AGT" && event == "s" ){
if (time < startTime){
startTime = time;
}
}
# Update total received packet size and store packets arrival time, to finally get the
# end time of the simulation
if (level == "AGT" && event == "r" ){
if (time > stopTime){
stopTime = time;
}
recvdSize += pkt_size;
}
}
END{
# calculate the throughput
printf("Average Throughput[kbps] = %.2f\n",(recvdSize/(stopTime-startTime)))
}
```

**Output:**


vesit510-08@vesit51008-HP-280-G2-MT: ~
File Edit View Search Terminal Help

```
vesit510-08@vesit51008-HP-280-G2-MT:~$ gawk -f throughput1.awk simple-dsdv.tr
Average Throughput[kbps] = 50450.46
vesit510-08@vesit51008-HP-280-G2-MT:~$ 
```

### 3. Getting Packet Delivery Ratio:

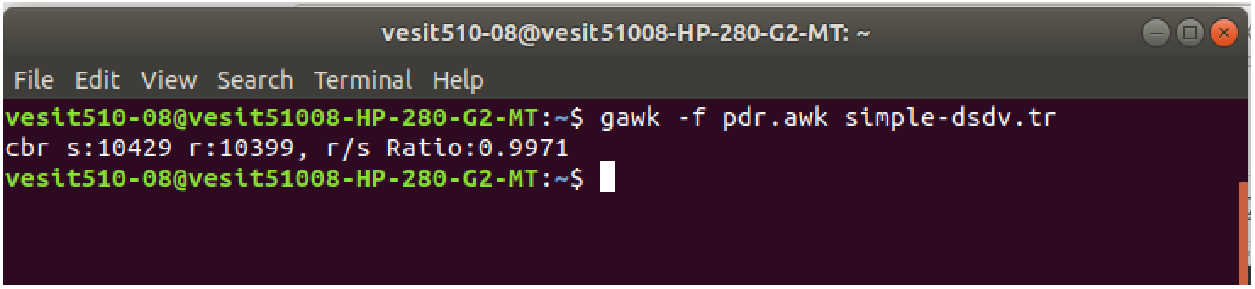
**Code:**

```
BEGIN {

sendLine = 0;
recvLine = 0;
}
$0 ~/^s.* AGT/ {
sendLine ++ ;
}
```

```
$0 ~/^r.* AGT/ {
recvLine ++ ;
}
END {
printf "cbr s:%d r:%d, r/s Ratio:%.4f \n", sendLine, recvLine,
(recvLine/sendLine);
}
```

**Output:**


vesit510-08@vesit51008-HP-280-G2-MT: ~
File Edit View Search Terminal Help

```
vesit510-08@vesit51008-HP-280-G2-MT:~$ gawk -f pdr.awk simple-dsdv.tr
cbr s:10429 r:10399, r/s Ratio:0.9971
vesit510-08@vesit51008-HP-280-G2-MT:~$
```

4. **Getting the number of packets dropped**

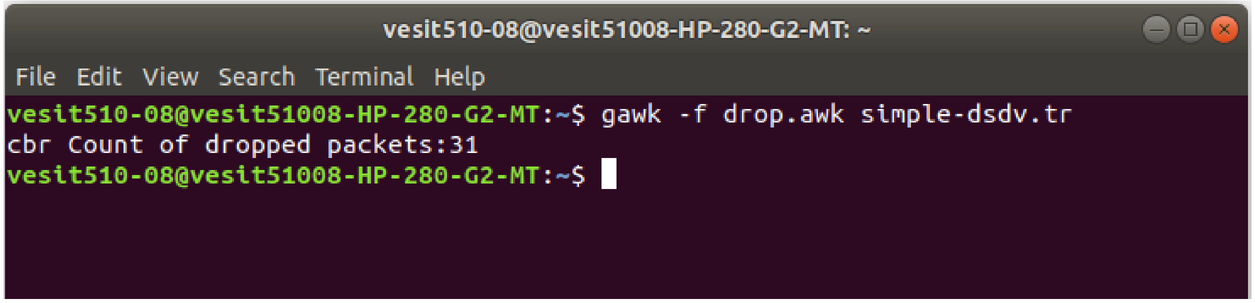
**Code:**

```
BEGIN{

countDropped = 0 ;
}
$0~/^d/{
countDropped++;
}
END{
printf"cbr Count of dropped packets:%d\n",countDropped;
}
```

Manav_17

## Output:


vesit510-08@vesit51008-HP-280-G2-MT: ~
File Edit View Search Terminal Help

```
vesit510-08@vesit51008-HP-280-G2-MT:~$ gawk -f drop.awk simple-dsdv.tr
cbr Count of dropped packets:31
vesit510-08@vesit51008-HP-280-G2-MT:~$ 
```

## Conclusion:

We have successfully understood how wireless propagation works and performed the aim of the experiment.