Program 1

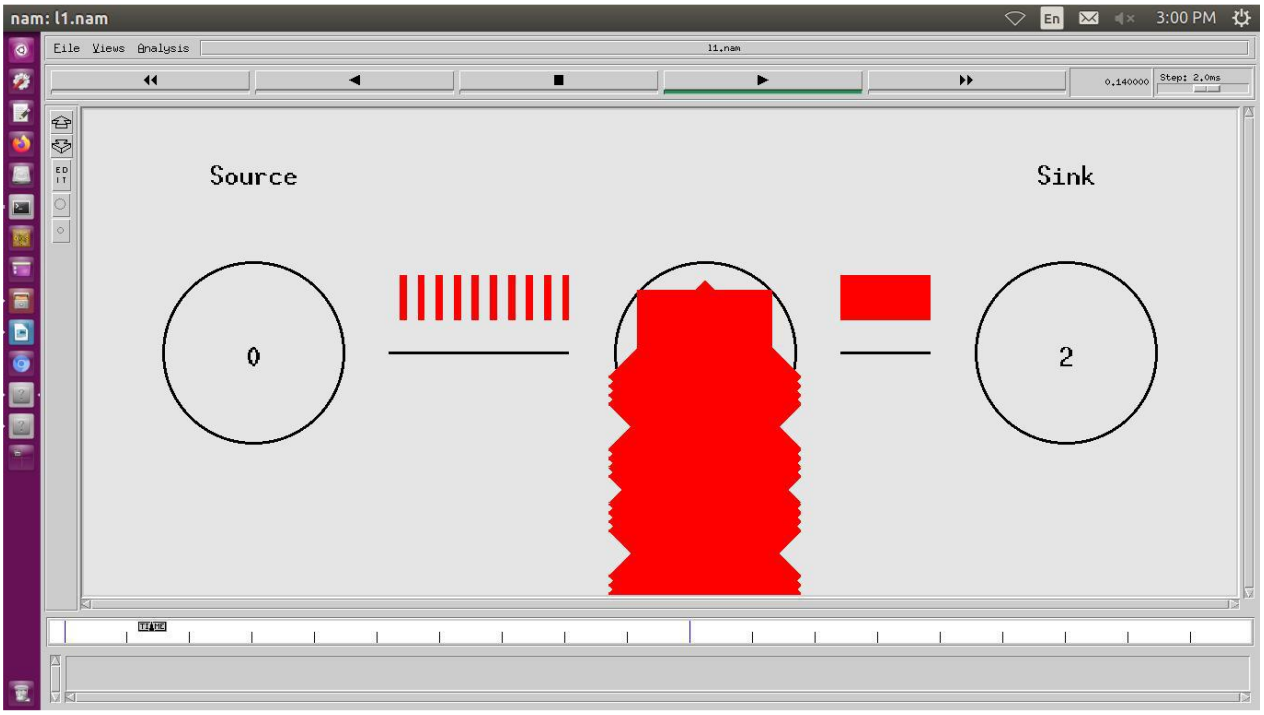
L1.tcl:-

```
set ns [new Simulator]
set f [open l1.tr w]
$ns trace-all $f
set nf [open l1.nam w]
$ns namtrace-all $nf
proc finish {} {
global f nf ns
$ns flush-trace
close $f
close $nf
exec nam l1.nam &
exit 0
}
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
$n0 label "Source"
$n2 label "Sink"
$ns color 1 red
$ns color 2 yellow
$ns duplex-link $n0 $n1 100Mb 10ms DropTail
$ns duplex-link $n1 $n2 10Mb 5ms DropTail
$ns queue-limit $n1 $n2 10
$ns duplex-link-op $n0 $n1 orient right
$ns duplex-link-op $n1 $n2 orient right
set udp0 [new Agent/UDP]
$ns attach-agent $n0 $udp0
set n [new Agent/Null]
$ns attach-agent $n2 $n
set cbr0 [new Application/Traffic/CBR]
$cbr0 attach-agent $udp0
$cbr0 set packetSize_ 5000
$cbr0 set interval_ 0.001
```

```
$ns connect $udp0 $n
$udp0 set class_ 1
$ns at 0.1 "$cbr0 start"
$ns at 1.8 "$cbr0 stop"
$ns at 2.0 "finish"
$ns run
```

L1.awk:-

```
BEGIN {
c=0;
}{
if($1=="d")
{
c++;
}
}
END {
printf("The number of packets dropped=%d¥n" , c);
}
```



PROGRAM 2

L2.tcl:-

```
set ns [ new Simulator ]
set nf [ open l2.nam w ]
$ns namtrace-all $nf
set tf [ open l2.tr w ]
$ns trace-all $tf
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
$ns duplex-link $n0 $n4 1005Mb 1ms DropTail
$ns duplex-link $n1 $n4 50Mb 1ms DropTail
$ns duplex-link $n2 $n4 2000Mb 1ms DropTail
$ns duplex-link $n3 $n4 200Mb 1ms DropTail
$ns duplex-link $n4 $n5 1Mb 1ms DropTail
set p0 [new Agent/Ping]
$ns attach-agent $n0 $p0
$p0 set packetSize_ 50000
$p0 set interval_ 0.0001
set p1 [new Agent/Ping]
$ns attach-agent $n1 $p1
set p2 [new Agent/Ping]
$ns attach-agent $n2 $p2
$p2 set packetSize_ 30000
$p2 set interval_ 0.00001
set p3 [new Agent/Ping]
$ns attach-agent $n3 $p3
set p5 [new Agent/Ping]
$ns attach-agent $n5 $p5
$ns queue-limit $n0 $n4 5
$ns queue-limit $n2 $n4 3
$ns queue-limit $n4 $n5 2
```

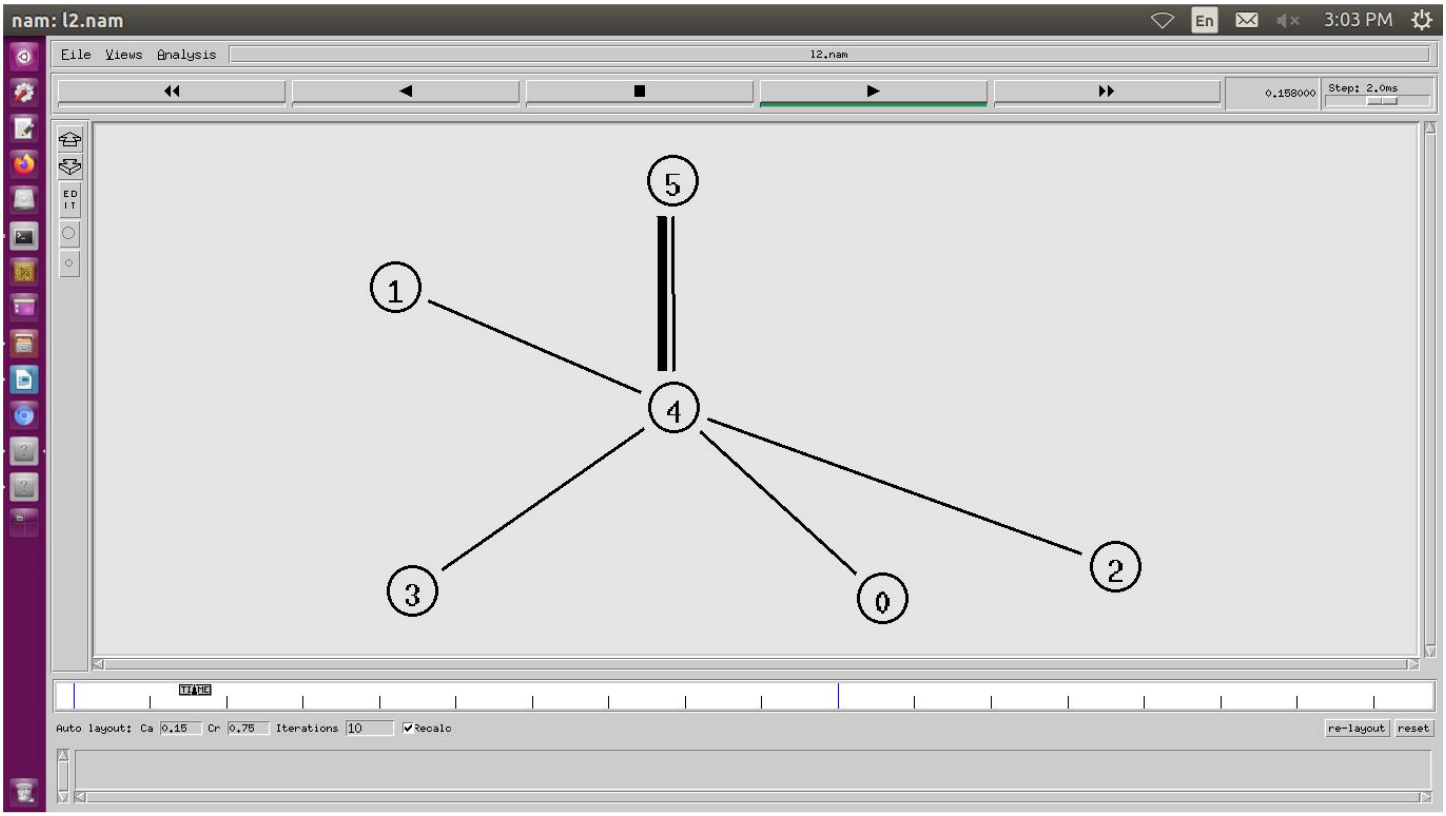
```
Agent/Ping instproc recv {from rtt} {
$self instvar node_
puts "node [$node_ id]received answer from $from with round trip time $rtt
msec"
}
$ns connect $p0 $p5
$ns connect $p2 $p3
proc finish { } {
global ns nf tf
$ns flush-trace
close $nf
close $tf
exec nam l2.nam &
exit 0
}
$ns at 0.1 "$p0 send"
$ns at 0.2 "$p0 send"
$ns at 0.3 "$p0 send"
$ns at 0.4 "$p0 send"
$ns at 0.5 "$p0 send"
$ns at 0.6 "$p0 send"
$ns at 0.7 "$p0 send"
$ns at 0.8 "$p0 send"
$ns at 0.9 "$p0 send"
$ns at 1.0 "$p0 send"
$ns at 0.1 "$p2 send"
$ns at 0.2 "$p2 send"
$ns at 0.3 "$p2 send"
$ns at 0.4 "$p2 send"
$ns at 0.5 "$p2 send"
$ns at 0.6 "$p2 send"
$ns at 0.7 "$p2 send"
$ns at 0.8 "$p2 send"
$ns at 0.9 "$p2 send"
$ns at 1.0 "$p2 send"
```

$ns at 2.0 "finish"

$ns run

L2.awk:-

```
BEGIN{
drop=0;
}{
if($1=="d" )
{
drop++;
}
}
END{
printf("Total number of %s packets dropped due to congestion=%d¥n",$5,drop);
}
```



PROGRAM 3 :-

L3.tcl:-

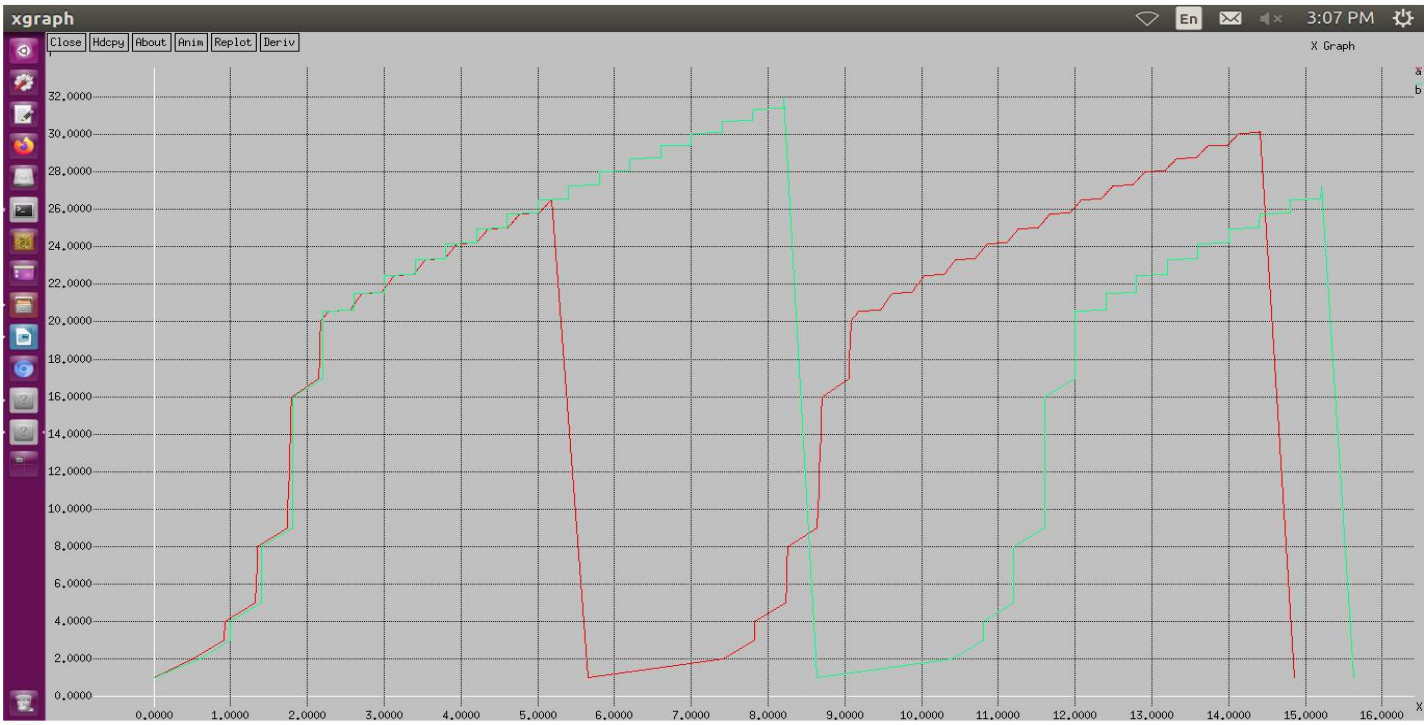
```
set ns [new Simulator]
set tf [open l3.tr w]
$ns trace-all $tf
set nf [open l3.nam w]
$ns namtrace-all $nf
set n0 [$ns node]
$n0 color "magenta"
$n0 label "src1"
set n1 [$ns node]
set n2 [$ns node]
$n2 color "magenta"
$n2 label "src2"
set n3 [$ns node]
$n3 color "blue"
$n3 label "dest2"
set n4 [$ns node]
set n5 [$ns node]
$n5 color "blue"
$n5 label "dest1"
$ns make-lan "$n0 $n1 $n2 $n3 $n4" 100Mb 100ms LL Queue/DropTail
Mac/802_3
$ns duplex-link $n4 $n5 1Mb 1ms DropTail
set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
$ftp0 set packetSize_ 500
$ftp0 set interval_ 0.0001
set sink5 [new Agent/TCPSink]
$ns attach-agent $n5 $sink5
$ns connect $tcp0 $sink5
set tcp2 [new Agent/TCP]
$ns attach-agent $n2 $tcp2
```

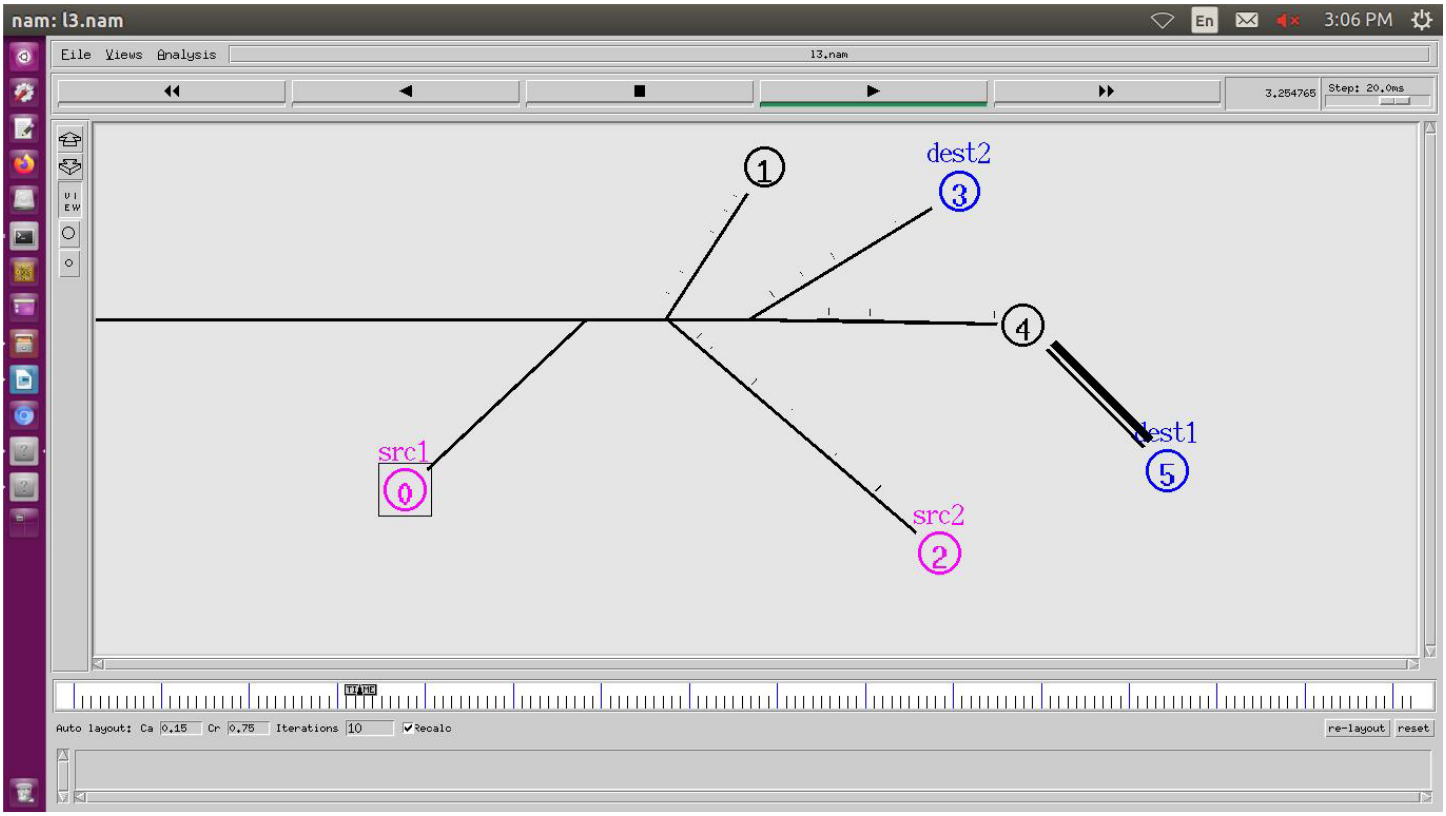
```
set ftp2 [new Application/FTP]
$ftp2 attach-agent $tcp2
$ftp2 set packetSize_ 600
$ftp2 set interval_ 0.001
set sink3 [new Agent/TCPSink]
$ns attach-agent $n3 $sink3
$ns connect $tcp2 $sink3
set file1 [open file1.tr w]
$tcp0 attach $file1
set file2 [open file2.tr w]
$tcp2 attach $file2
$tcp0 trace cwnd_
$tcp2 trace cwnd_
proc finish { } {
global ns nf tf
$ns flush-trace
close $tf
close $nf
exec nam l3.nam &
exit 0
}
$ns at 0.1 "$ftp0 start"
$ns at 5 "$ftp0 stop"
$ns at 7 "$ftp0 start"
$ns at 0.2 "$ftp2 start"
$ns at 8 "$ftp2 stop"
$ns at 14 "$ftp0 stop"
$ns at 10 "$ftp2 start"
$ns at 15 "$ftp2 stop"
$ns at 16 "finish"
$ns run
```

L3.awk:-

```
BEGIN {
}{ if($6
=="cwnd_")
```

```
printf("%f¥t%f¥t¥n",$1,$7);
}
END {
}
```





PROGRAM 4

L4.tcl:-

```
set ns [new Simulator]
set tf [open p4.tr w]
$ns trace-all $tf
set topo [new Topography]
$topo load_flatgrid 1000 1000
set nf [open p4.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 ¥
-phyType Phy/WirelessPhy ¥
-channelType Channel/WirelessChannel ¥
-propType 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
```

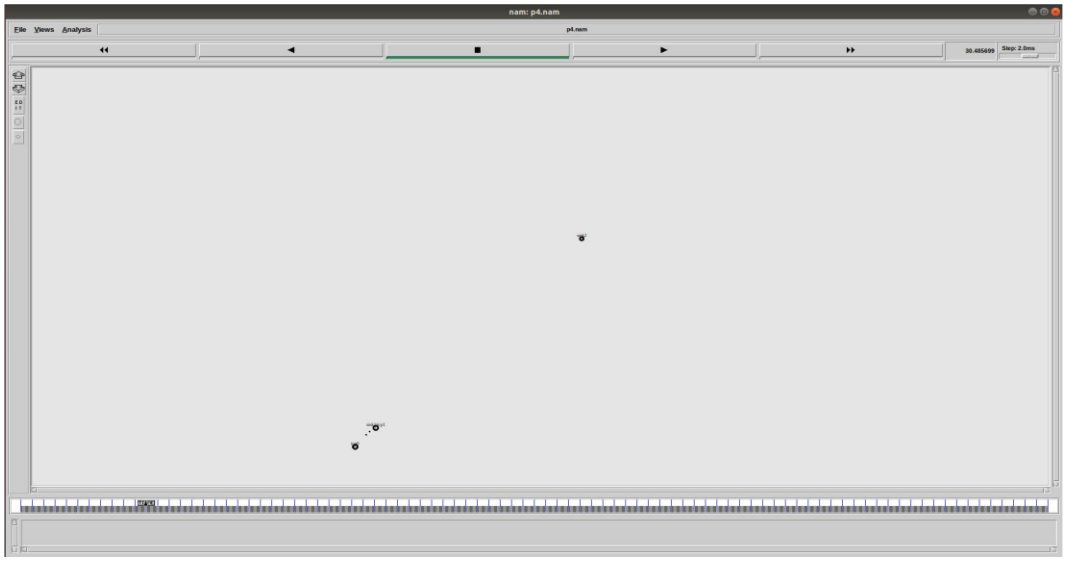
```
$n2 set X_ 600
$n2 set Y_ 600
$n2 set Z_ 0
$ns at 0.1 "$n0 setdest 50 50 15"
$ns at 0.1 "$n1 setdest 100 100 25"
$ns at 0.1 "$n2 setdest 600 600 25"
set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
set sink1 [new Agent/TCPSink]
$ns attach-agent $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 nf tf
$ns flush-trace
exec nam p4.nam &
close $tf
exit 0
}
$ns at 250 "finish"
$ns run
```

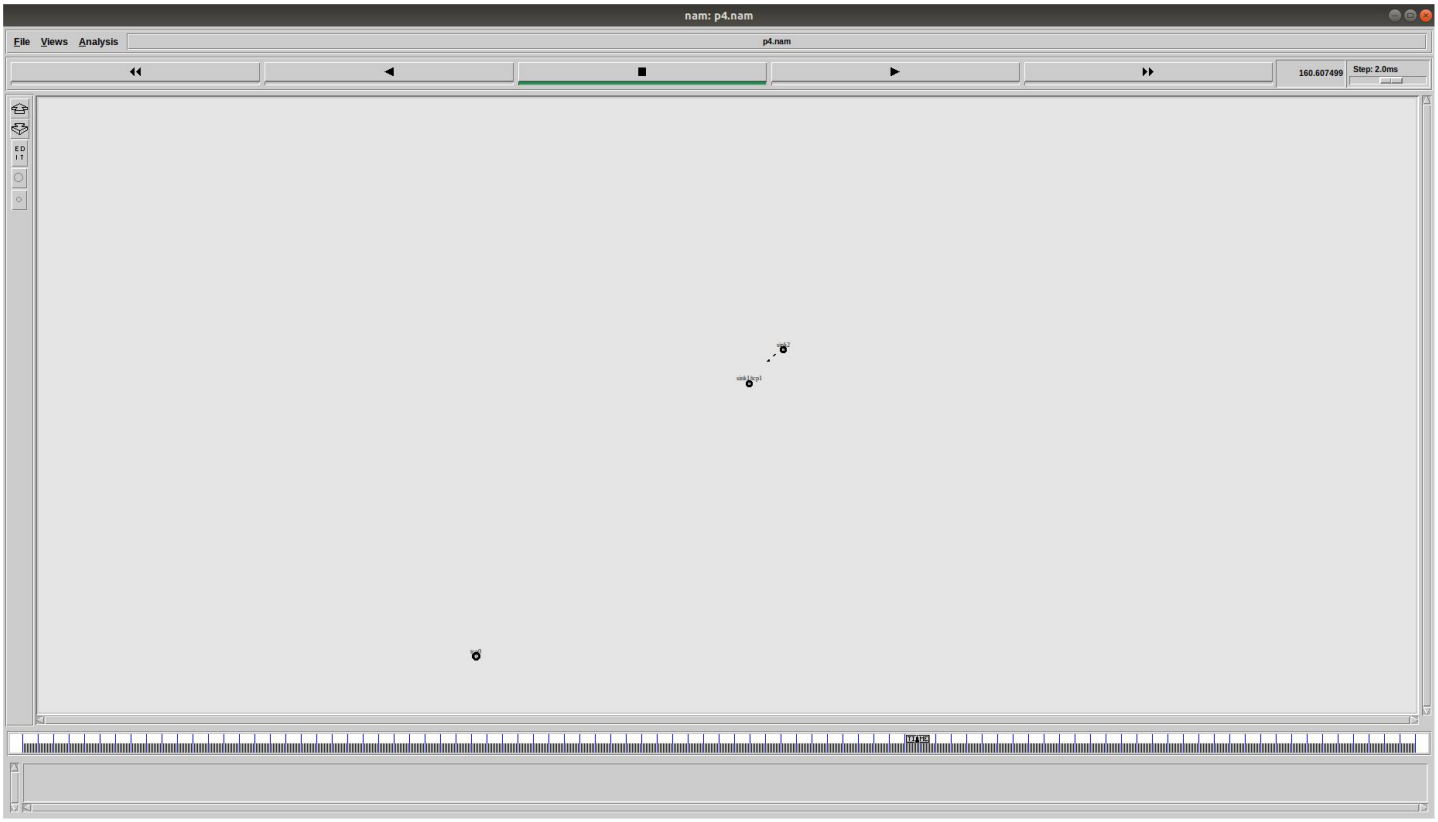
L4.awk:-

```
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 ¥n",
((count2*pack2*8)/(time2*1000000)));
}
```





PROGRAM 5:-

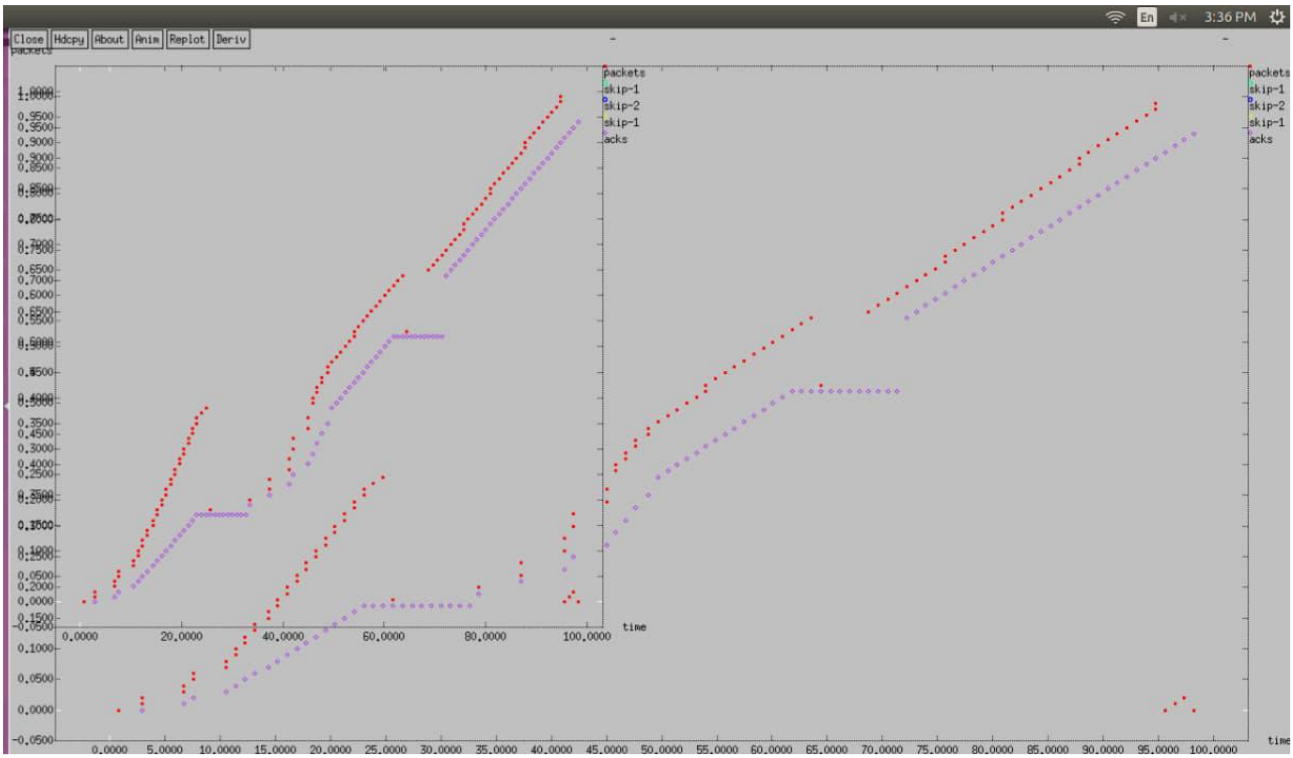
L5.tcl

```
# General Parameters
set stop 100 ;
# Stop time.
# Topology
set type gsm ;
#type of link:
# AQM parameters
set minth 30 ;
set maxth 0 ;
set adaptive 1 ;
# 1 for Adaptive RED, 0 for plain RED
# Traffic generation.
set flows 0 ;
# number of long-lived TCP flows
set window 30 ;
# window for long-lived traffic
set web 2 ;
# number of web sessions
```

```
# Plotting statics.
set opt(wrap) 100 ;
# wrap plots?
set opt(srcTrace) is ; # where to plot traffic
set opt(dstTrace) bs2 ;
# where to plot traffic
#default downlink bandwidth in bps
set bwDL(gsm) 9600
#default uplink bandwidth in bps
set bwUL(gsm) 9600
#default downlink propagation delay in seconds
set propDL(gsm) .500
#default uplink propagation delay in seconds
set propUL(gsm) .500
set ns [new Simulator]
set tf [open p5.tr w]
$ns trace-all $tf
set nodes(is) [$ns node]
set nodes(ms) [$ns node]
set nodes(bs1) [$ns node]
set nodes(bs2) [$ns node]
set nodes(lp) [$ns node]
proc cell_topo {} {
global ns nodes
$ns duplex-link $nodes(lp) $nodes(bs1) 3Mbps 10nodes(ms) DropTail
$ns duplex-link $nodes(bs1) $nodes(ms) 1 1 RED
$ns duplex-link $nodes(ms) $nodes(bs2) 1 1 RED
$ns duplex-link $nodes(bs2) $nodes(is) 3Mbps 50nodes(ms) DropTail
puts " GSM Cell Topology"
}
proc set_link_para {t} {
global ns nodes bwUL bwDL propUL propDL buf
$ns bandwidth $nodes(bs1) $nodes(ms) $bwDL($t) duplex
$ns bandwidth $nodes(bs2) $nodes(ms) $bwDL($t) duplex
$ns delay $nodes(bs1) $nodes(ms) $propDL($t) duplex
```

```
$ns delay $nodes(bs2) $nodes(ms) $propDL($t) duplex
$ns queue-limit $nodes(bs1) $nodes(ms) 10
$ns queue-limit $nodes(bs2) $nodes(ms) 10
}# RED and TCP parameters
Queue/RED set adaptive_ $adaptive
Queue/RED set thresh_ $minth
Queue/RED set maxthresh_ $maxth
Agent/TCP set window_ $window
source web.tcl
#Create topology
switch $type {
gsm -
gprs -
umts {cell_topo}
}
set_link_para $type
$ns insert-delayer $nodes(ms) $nodes(bs1) [new Delayer]
$ns insert-delayer $nodes(ms) $nodes(bs2) [new Delayer]
# Set up forward TCP connection
if {$flows == 0} {
set tcp1 [$ns create-connection TCP/Sack1 $nodes(is) TCPSink/Sack1
$nodes(lp) 0]
set ftp1 [[set tcp1] attach-app FTP]
$ns at 0.8 "[set ftp1] start"
} if {$flows >
0} {
set tcp1 [$ns create-connection TCP/Sack1 $nodes(is) TCPSink/Sack1
$nodes(lp) 0]
set ftp1 [[set tcp1] attach-app FTP]
$tcp1 set window_ 100
$ns at 0.0 "[set ftp1] start"
$ns at 3.5 "[set ftp1] stop"
set tcp2 [$ns create-connection TCP/Sack1 $nodes(is) TCPSink/Sack1
$nodes(lp) 0]
set ftp2 [[set tcp2] attach-app FTP]
```

```
$tcp2 set window_ 3
$ns at 1.0 "[set ftp2] start"
$ns at 8.0 "[set ftp2] stop"
}
proc stop {} {
global nodes opt nf
set wrap $opt(wrap)
set sid [$nodes($opt(srcTrace)) id]
set did [$nodes($opt(dstTrace)) id]
set a "out.tr"
set GETRC "../../../bin/getrc"
set RAW2XG "../../../bin/raw2xg"
exec $GETRC -s $sid -d $did -f 0 p5.tr | ¥
$RAW2XG -s 0.01 -m $wrap -r > plot.xgr
exec $GETRC -s $did -d $sid -f 0 p5.tr | ¥$RAW2XG -a -s 0.01 -m $wrap >>
plot.xgr
exec xgraph -x time -y packets plot.xgr &
exit 0
}
$ns at $stop "stop"
$ns run
```



PROGRAM 6:-

```
set opt(title) zero ;
set opt(stop) 100 ;
set opt(ecn) 0
;
set opt(type) umts ;
set opt(secondDelay) 55 ;
set opt(minth) 30 ;
set opt(maxth) 0 ;
set opt(adaptive) 1 ;
set opt(flows) 0 ;
set opt(window) 30 ;
set opt(web) 2
;
set opt(quiet) 0 ;
set opt(wrap) 100 ;
set opt(srcTrace) is ;
set opt(dstTrace) bs2 ;
set opt(umtsbuf) 10 ;
set bwDL(umts) 384000
set bwUL(umts) 64000
set propDL(umts) .150
set propUL(umts) .150
set buf(umts) 20
set ns [new Simulator]
set tf [open l6.tr w]
$ns trace-all $tf
set nodes(is) [$ns node]
set nodes(ms) [$ns node]
set nodes(bs1) [$ns node]
set nodes(bs2) [$ns node]
set nodes(lp) [$ns node]
proc cell_topo {} {
global ns nodes
$ns duplex-link $nodes(lp) $nodes(bs1) 3Mbps 10ms DropTail
```

```
$ns duplex-link $nodes(bs1) $nodes(ms) 1 1 RED
$ns duplex-link $nodes(ms) $nodes(bs2) 1 1 RED
$ns duplex-link $nodes(bs2) $nodes(is) 3Mbps 50ms DropTail
puts "Cell Topology"
}
proc set_link_params {t} {
global ns nodes bwUL bwDL propUL propDL buf
$ns bandwidth $nodes(bs1) $nodes(ms) $bwDL($t) simplex
$ns bandwidth $nodes(ms) $nodes(bs1) $bwUL($t) simplex
$ns delay $nodes(bs1) $nodes(ms) $propDL($t) simplex
$ns delay $nodes(ms) $nodes(bs1) $propDL($t) simplex $ns queue-limit
$nodes(bs1) $nodes(ms) $buf($t)
$ns queue-limit $nodes(ms) $nodes(bs1) $buf($t)
$ns bandwidth $nodes(bs2) $nodes(ms) $bwDL($t) simplex
$ns bandwidth $nodes(ms) $nodes(bs2) $bwUL($t) simplex
$ns delay $nodes(bs2) $nodes(ms) $propDL($t) simplex
$ns delay $nodes(ms) $nodes(bs2) $propDL($t) simplex
$ns queue-limit $nodes(bs2) $nodes(ms) $buf($t)
$ns queue-limit $nodes(ms) $nodes(bs2) $buf($t)
}
Queue/RED set summarystats_ true
Queue/DropTail set summarystats_ true
Queue/RED set adaptive_ $opt(adaptive)
Queue/RED set q_weight_ 0.0
Queue/RED set thresh_ $opt(minth)
Queue/RED set maxthresh_ $opt(maxth)
Queue/DropTail set shrink_drops_ true
Agent/TCP set ecn_ $opt(ecn)
Agent/TCP set window_ $opt(window)
DelayLink set avoidReordering_ true
source web.tcl
switch $opt(type) {
umts {cell_topo}
}
set_link_params $opt(type)
```

```
$ns insert-delayer $nodes(ms) $nodes(bs1) [new Delayer]
$ns insert-delayer $nodes(bs1) $nodes(ms) [new Delayer]
$ns insert-delayer $nodes(ms) $nodes(bs2) [new Delayer]
$ns insert-delayer $nodes(bs2) $nodes(ms) [new Delayer]
if {$opt(flows) == 0} {
set tcp1 [$ns create-connection TCP/Sack1 $nodes(is) TCPSink/Sack1
$nodes(lp) 0]
set ftp1 [[set tcp1] attach-app FTP]
$ns at 0.8 "[set ftp1] start"
} if {$opt(flows) >
0} {
set tcp1 [$ns create-connection TCP/Sack1 $nodes(is) TCPSink/Sack1
$nodes(lp) 0]
set ftp1 [[set tcp1] attach-app FTP]
$tcp1 set window_ 100
$ns at 0.0 "[set ftp1] start"
$ns at 3.5 "[set ftp1] stop"
set tcp2 [$ns create-connection TCP/Sack1 $nodes(is) TCPSink/Sack1
$nodes(lp) 0]
set ftp2 [[set tcp2] attach-app FTP]
$tcp2 set window_ 3
$ns at 1.0 "[set ftp2] start"
$ns at 8.0 "[set ftp2] stop"
}
proc stop {} {
global nodes opt nf
set wrap $opt(wrap)
set sid [$nodes($opt(srcTrace)) id]
set did [$nodes($opt(dstTrace)) id]
if {$opt(srcTrace) == "is"} {
set a "-a out.tr"
} else {
set a "out.tr"
}
set GETRC "../../../bin/getrc"
```

```
set RAW2XG "../../../bin/raw2xg"
exec $GETRC -s $sid -d $did -f 0 l6.tr | ¥
$RAW2XG -s 0.01 -m $wrap -r > plot.xgr
exec $GETRC -s $did -d $sid -f 0 l6.tr | ¥
$RAW2XG -a -s 0.01 -m $wrap >> plot.xgr
exec $GETRC -s $sid -d $did -f 1 l6.tr | ¥
$RAW2XG -s 0.01 -m $wrap -r >> plot.xgr
exec $GETRC -s $did -d $sid -f 1 l6.tr | ¥
$RAW2XG -s 0.01 -m $wrap -a >> plot.xgr
exec ./xg2gp.awk plot.xgr
if {!$opt(quiet)} {
exec xgraph -bb -tk -nl -m -x time -y packets plot.xgr &
}
exit 0
}
$ns at $opt(stop) "stop"
$ns run
```