PART A

1. Implement three nodes point – to – point network with duplex links between them. Set the queue size, vary the bandwidth and find the number of packets dropped.

```
set ns [new Simulator]             /* Letter S is capital */

set nf [open lab1.nam w]           /* open a nam trace file in write mode */
$ns namtrace-all $nf               /* nf – nam file */

set tf [open lab1.tr w]            /* tf- trace file */
$ns trace-all $tf

proc finish { } {                  /* provide space b/w proc and finish and all are in small case */
global ns nf tf
$ns flush-trace                    /* clears trace file contents */
close $nf
close $tf
exec nam lab1.nam &
exit 0
}

set n0 [$ns node]                  /* creates 4 nodes */
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]

$ns duplex-link $n0 $n2 200Mb 10ms DropTail          /*Letter M is capital Mb*/
$ns duplex-link $n1 $n2 100Mb 5ms DropTail           /*D and T are capital*/
$ns duplex-link $n2 $n3 1Mb 1000ms DropTail

$ns queue-limit $n0 $n2 10
$ns queue-limit $n1 $n2 10

set udp0 [new Agent/UDP]                         /* Letters A,U,D and P are capital */
$ns attach-agent $n0 $udp0

set cbr0 [new Application/Traffic/CBR]           /* A,T,C,B and R are capital*/
$cbr0 set packetSize_ 500                        /*S is capital, space after underscore*/
$cbr0 set interval_ 0.005
$cbr0 attach-agent $udp0

set udp1 [new Agent/UDP]
$ns attach-agent $n1 $udp1
set cbr1 [new Application/Traffic/CBR]
$cbr1 attach-agent $udp1

set udp2 [new Agent/UDP]
$ns attach-agent $n2 $udp2
```

```
set cbr2 [new Application/Traffic/CBR]
$cbr2 attach-agent $udp2

set null0 [new Agent/Null]                                  /* A and N are capital */
$ns attach-agent $n3 $null0

$ns connect $udp0 $null0
$ns connect $udp1 $null0

$ns at 0.1 "$cbr0 start"
$ns at 0.2 "$cbr1 start"

$ns at 1.0 "finish"
$ns run
```

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

/*immediately after BEGIN should open braces '{'

```
BEGIN {
c=0;
}
{
If ($1= ="d")
{
c++;
printf("%s\t%s\n",$5,$11);
}
}
```

/*immediately after END should open braces '{' END{
printf("The number of packets dropped
=%d\n",c); }

Steps for execution

1) Open gedit and type the program. program name should have extension **".tcl" [root@localhost ~]# gedit lab1.tcl**
2) Save the program by pressing **"CTRL + S"**
3) Open gedit and type awk program. program name should have extension **".awk" [root@localhost ~]# gedit lab1.awk**
4) Save the program by pressing **"CTRL + S"**
5)Run the simulation program

[root@localhost~]# ns lab1.tcl

i) Here **"ns"** indicates network simulator. We get the topology shown in the snapshot.

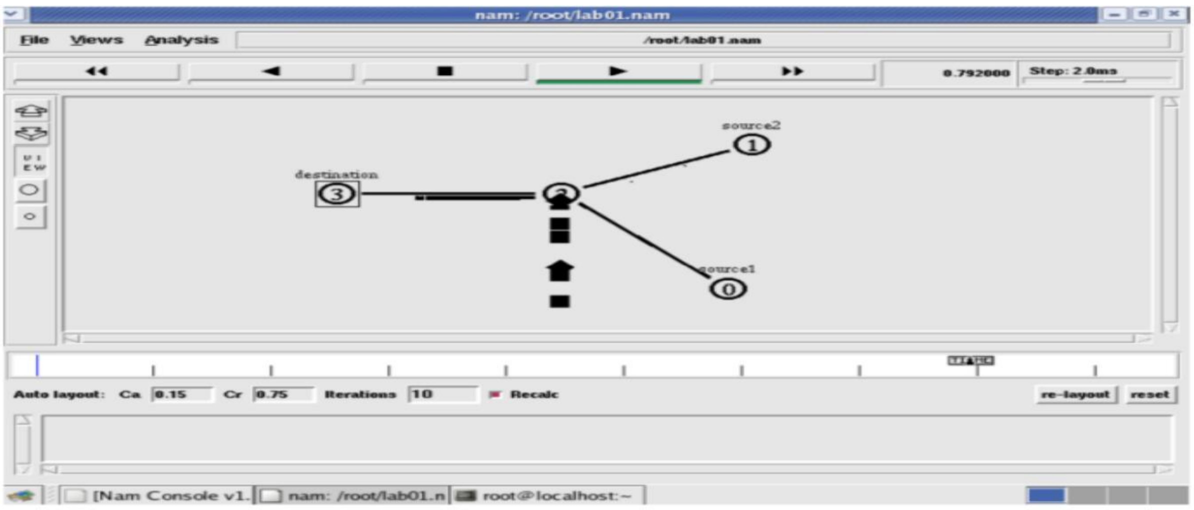
ii) Now press the play button in the simulation window and the simulation will begins.

6) After simulation is completed run awk file to see the
output , [root@localhost~]# **awk –f lab1.awk lab1.tr**

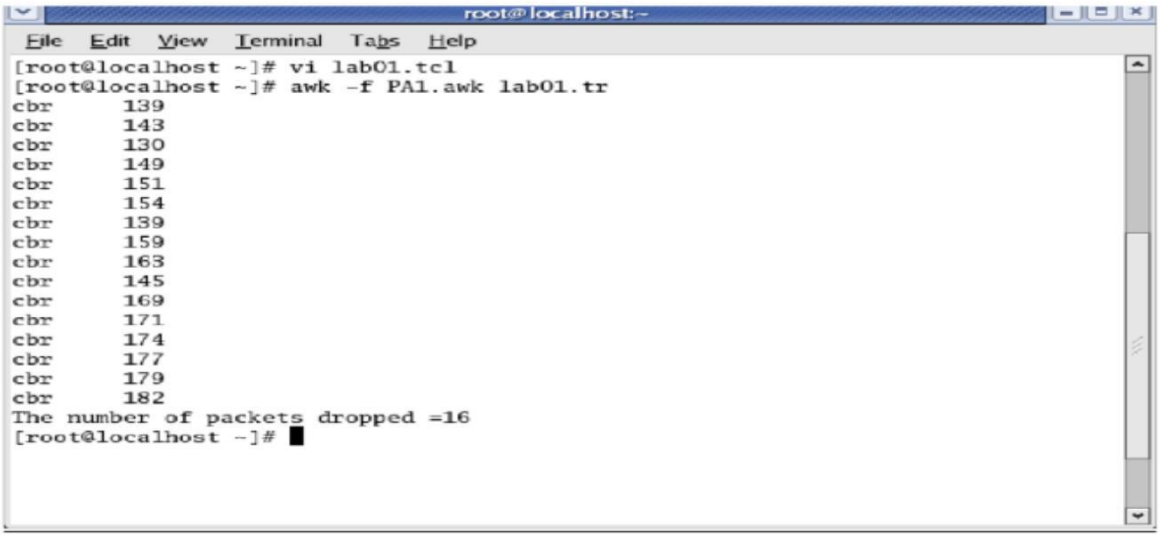
7) To see the trace file contents open the
file as , [root@localhost~]# **vi lab1.tr**

Trace file contains 12 columns:-
Event type, Event time, From Node, Source Node, Packet Type, Packet Size, Flags(indicated by --------), Flow ID, Source address, Destination address, Sequence ID,Packet ID

Topology



Output



```
[root@localhost ~]# vi lab01.tcl
[root@localhost ~]# awk -f PA1.awk lab01.tr
cbr       139
cbr       143
cbr       130
cbr       149
cbr       151
cbr       154
cbr       139
cbr       159
cbr       163
cbr       145
cbr       169
cbr       171
cbr       174
cbr       177
cbr       179
cbr       182
The number of packets dropped =16
[root@localhost ~]# 
```

Note:

1. Set the queue size fixed from n0 to n2 as 10, n1-n2 to 10 and from n2-n3 as 5.
   Syntax: To set the queue size
   $ns set queue-limit <from> <to> <size> Eg:
   $ns set queue-limit $n0 $n2 10
2. Go on varying the bandwidth from 10, 20 30 . . and find the number of packets dropped at the node 2

2. Implement transmission of ping messages/trace route over a network topology consisting of 6 nodes and find the number of packets dropped due to congestion.

```
set ns [ new Simulator ]

set nf [ open lab2.nam w ]
$ns namtrace-all $nf

set tf [ open lab2.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]

$n4 shape box

$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 p1 [new Agent/Ping]
$ns attach-agent $n0 $p1
$p1 set packetSize_ 50000
$p1 set interval_ 0.0001

set p2 [new Agent/Ping]
$ns attach-agent $n1 $p2

set p3 [new Agent/Ping]
$ns attach-agent $n2 $p3
$p3 set packetSize_ 30000
$p3 set interval_ 0.00001

set p4 [new Agent/Ping]
$ns attach-agent $n3 $p4

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" }

# please provide space between $node_ and id. No space between $ and
from. No #space between and $ and rtt */

$ns connect $p1 $p5
$ns connect $p3 $p4

proc finish { } {
global ns nf tf
$ns flush-trace
close $nf
close $tf
exec nam lab2.nam &
exit 0
}


$ns at 0.1 "$p1 send"
$ns at 0.2 "$p1 send"
$ns at 0.3 "$p1 send"
$ns at 0.4 "$p1 send"
$ns at 0.5 "$p1 send"
$ns at 0.6 "$p1 send"
$ns at 0.7 "$p1 send"
$ns at 0.8 "$p1 send"
$ns at 0.9 "$p1 send"
$ns at 1.0 "$p1 send"
$ns at 1.1 "$p1 send"
$ns at 1.2 "$p1 send"
$ns at 1.3 "$p1 send"
$ns at 1.4 "$p1 send"
$ns at 1.5 "$p1 send"
$ns at 1.6 "$p1 send"
$ns at 1.7 "$p1 send"
$ns at 1.8 "$p1 send"
$ns at 1.9 "$p1 send"
$ns at 2.0 "$p1 send"
$ns at 2.1 "$p1 send"
$ns at 2.2 "$p1 send"
$ns at 2.3 "$p1 send"
$ns at 2.4 "$p1 send"
$ns at 2.5 "$p1 send"
$ns at 2.6 "$p1 send"
$ns at 2.7 "$p1 send"
$ns at 2.8 "$p1 send"
$ns at 2.9 "$p1 send"
```

```
$ns at 0.1 "$p3 send"
$ns at 0.2 "$p3 send"
$ns at 0.3 "$p3 send"
$ns at 0.4 "$p3 send"
$ns at 0.5 "$p3 send"
$ns at 0.6 "$p3 send"
$ns at 0.7 "$p3 send"
$ns at 0.8 "$p3 send"
$ns at 0.9 "$p3 send"
$ns at 1.0 "$p3 send"
$ns at 1.1 "$p3 send"
$ns at 1.2 "$p3 send"
$ns at 1.3 "$p3 send"
$ns at 1.4 "$p3 send"
$ns at 1.5 "$p3 send"
$ns at 1.6 "$p3 send"
$ns at 1.7 "$p3 send"
$ns at 1.8 "$p3 send"
$ns at 1.9 "$p3 send"
$ns at 2.0 "$p3 send"
$ns at 2.1 "$p3 send"
$ns at 2.2 "$p3 send"
$ns at 2.3 "$p3 send"
$ns at 2.4 "$p3 send"
$ns at 2.5 "$p3 send"
$ns at 2.6 "$p3 send"
$ns at 2.7 "$p3 send"
$ns at 2.8 "$p3 send"
$ns at 2.9 "$p3 send"

$ns at 3.0 "finish"

$ns run
```

AWK file (Open a new editor using “vi command” and write awk file and save with “.awk” extension)

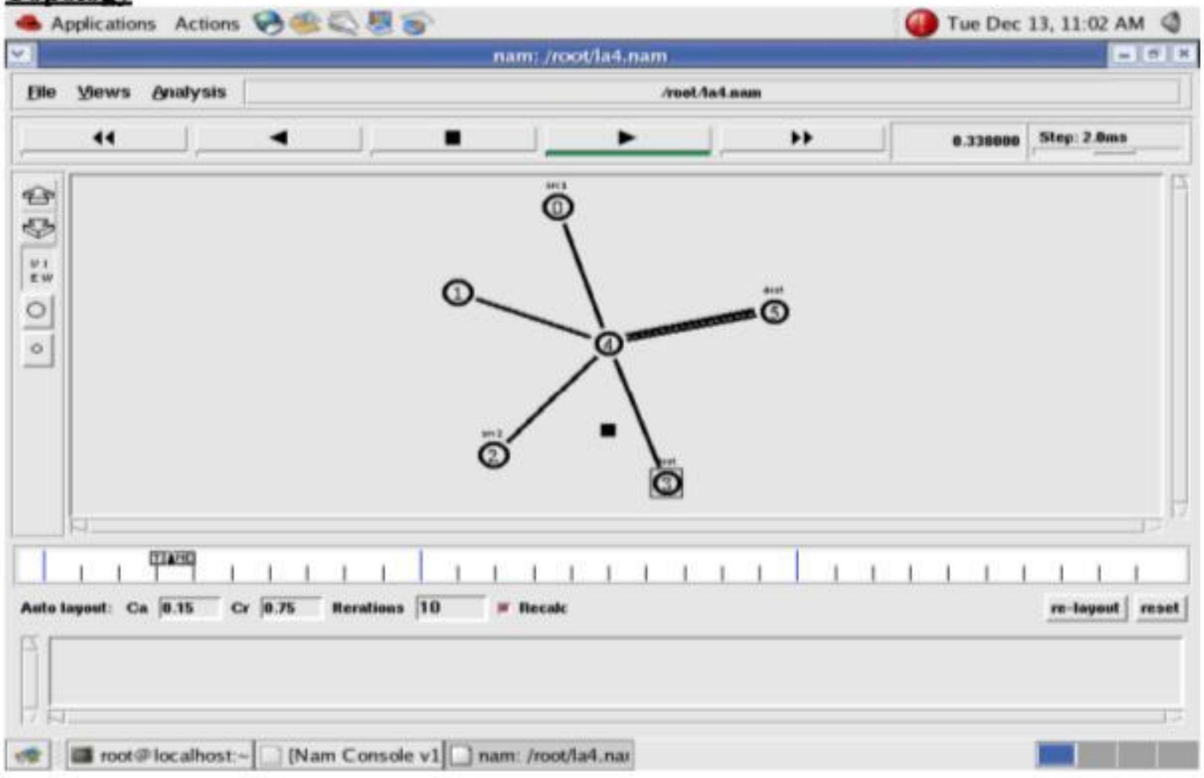
```
BEGIN{
drop=0;
}
{
if($1= ="d" )
{
drop++;
}
}
END{
printf("Total number of %s packets dropped due to congestion =%d\n",$5,drop);
}
```

Steps for execution

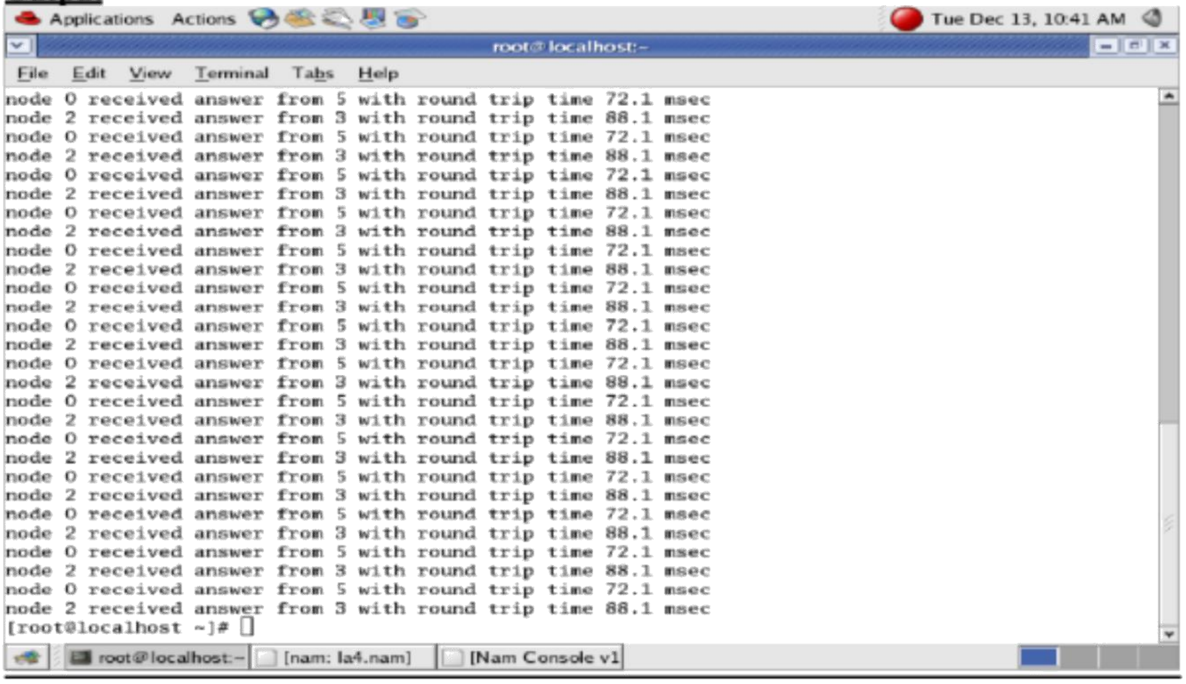
1) Open gedit and type the program. program name should have extension **“.tcl” [root@localhost ~]# gedit lab2.tcl**
2) Save the program by pressing **“CTRL + S”**
3) Open gedit and type awk program. program name should have extension **“.awk” [root@localhost ~]# gedit lab2.awk**
4) Save the program by pressing **“CTRL + S”**
5)Run the simulation program
   [root@localhost~]# ns lab2.tcl
i) Here **“ns”** indicates network simulator. We get the topology shown in the snapshot.
ii) Now press the play button in the simulation window and the simulation will begins.

6) After simulation is completed run awk file to see the output , [root@localhost~]# **awk –f lab2.awk lab2.tr**
7) To see the trace file contents open the file as , [root@localhost~]# **vi lab2.tr**

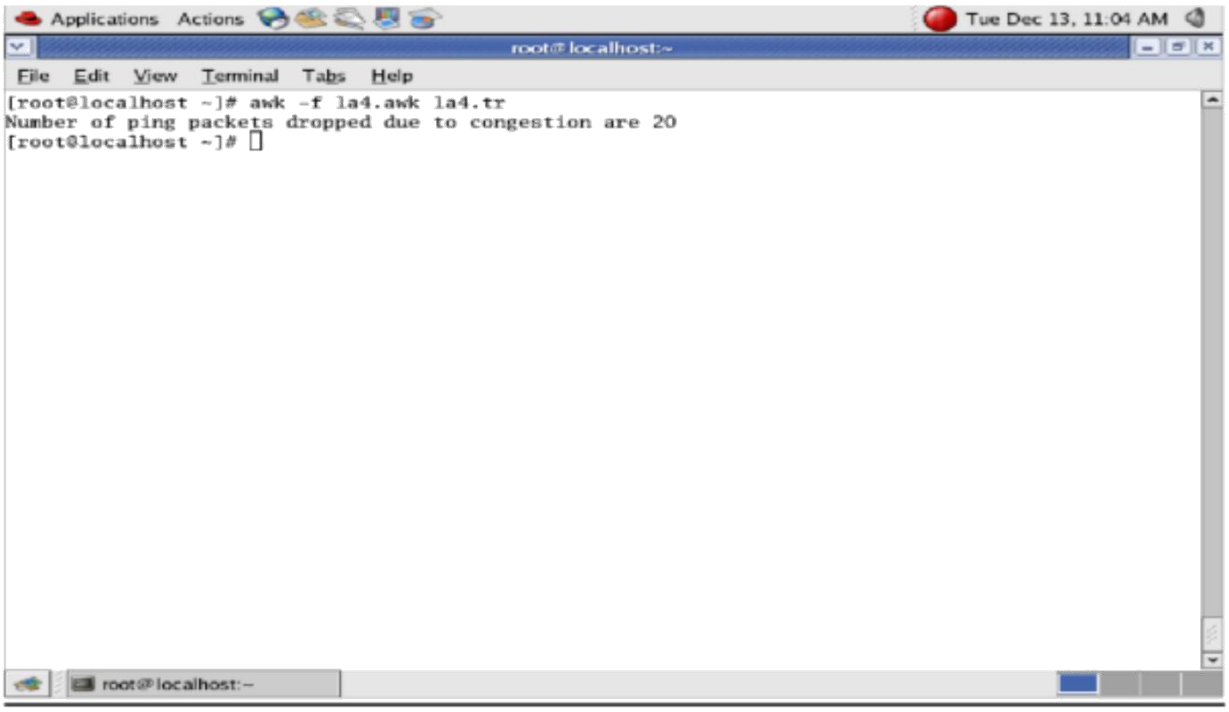
Topology



Output



```
node 0 received answer from 5 with round trip time 72.1 msec
node 2 received answer from 3 with round trip time 88.1 msec
node 0 received answer from 5 with round trip time 72.1 msec
node 2 received answer from 3 with round trip time 88.1 msec
node 0 received answer from 5 with round trip time 72.1 msec
node 2 received answer from 3 with round trip time 88.1 msec
node 0 received answer from 5 with round trip time 72.1 msec
node 2 received answer from 3 with round trip time 88.1 msec
node 0 received answer from 5 with round trip time 72.1 msec
node 2 received answer from 3 with round trip time 88.1 msec
node 0 received answer from 5 with round trip time 72.1 msec
node 2 received answer from 3 with round trip time 88.1 msec
node 0 received answer from 5 with round trip time 72.1 msec
node 2 received answer from 3 with round trip time 88.1 msec
node 0 received answer from 5 with round trip time 72.1 msec
node 2 received answer from 3 with round trip time 88.1 msec
node 0 received answer from 5 with round trip time 72.1 msec
node 2 received answer from 3 with round trip time 88.1 msec
node 0 received answer from 5 with round trip time 72.1 msec
node 2 received answer from 3 with round trip time 88.1 msec
node 0 received answer from 5 with round trip time 72.1 msec
node 2 received answer from 3 with round trip time 88.1 msec
node 0 received answer from 5 with round trip time 72.1 msec
node 2 received answer from 3 with round trip time 88.1 msec
node 0 received answer from 5 with round trip time 72.1 msec
node 2 received answer from 3 with round trip time 88.1 msec
node 0 received answer from 5 with round trip time 72.1 msec
node 2 received answer from 3 with round trip time 88.1 msec
[root@localhost ~]#
```



```
[root@localhost ~]# awk -f la4.awk la4.tr
Number of ping packets dropped due to congestion are 20
[root@localhost ~]#
```

Note:

Vary the bandwidth and queue size between the nodes n0-n2 , n2-n4. n6-n2 and n2- n5 and see the number of packets dropped at the nodes.

3. Implement an Ethernet LAN using n nodes and set multiple traffic nodes and plot congestion window for different source / destination.

```
set ns [new Simulator]
set tf [open lab3.tr w]
$ns trace-all $tf

set nf [open lab3.nam w]
$ns namtrace-all $nf

set n0 [$ns node]
$n0 color "pink"
$n0 label "src1"

set n1 [$ns node]
$n1 color "red"

set n2 [$ns node]
$n2 color "pink"
$n2 label "src2"

set n3 [$ns node]
$n3 color "blue"
$n3 label "dest2"

set n4 [$ns node]
$n4 shape square

set n5 [$ns node]
$n5 color "blue"
$n5 label "dest1"

$ns make-lan "$n0 $n1 $n2 $n3 $n4 " 50Mb 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 sink0 [new Agent/TCPSink]
$ns attach-agent $n5 $sink0
$ns connect $tcp0 $sink0

set tcp1 [new Agent/TCP]
$ns attach-agent $n2 $tcp1

set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1
$ftp1 set packetSize_ 600
$ftp1 set interval_ 0.001

set sink1 [new Agent/TCPSink]
$ns attach-agent $n3 $sink1
$ns connect $tcp1 $sink1

set file1 [open file1.tr w]
$tcp0 attach $file1

set file2 [open file2.tr w]
$tcp1 attach $file2

$tcp0 trace cwnd_
$tcp1 trace cwnd_

proc finish { } {
global ns nf tf
$ns flush-trace
close $tf
close $nf
exec nam lab3.nam &
exit 0
}

$ns at 0.1 "$ftp0 start"
$ns at 5 "$ftp0 stop"
$ns at 7 "$ftp0 start"
$ns at 0.2 "$ftp1 start"
$ns at 8 "$ftp1 stop"
$ns at 14 "$ftp0 stop"
$ns at 10 "$ftp1 start"
$ns at 15 "$ftp1 stop"
$ns at 16 "finish"
$ns run
```

AWK file (Open a new editor using “vi command” and write awk file and save with “.awk” extension)

cwnd:- means congestion window

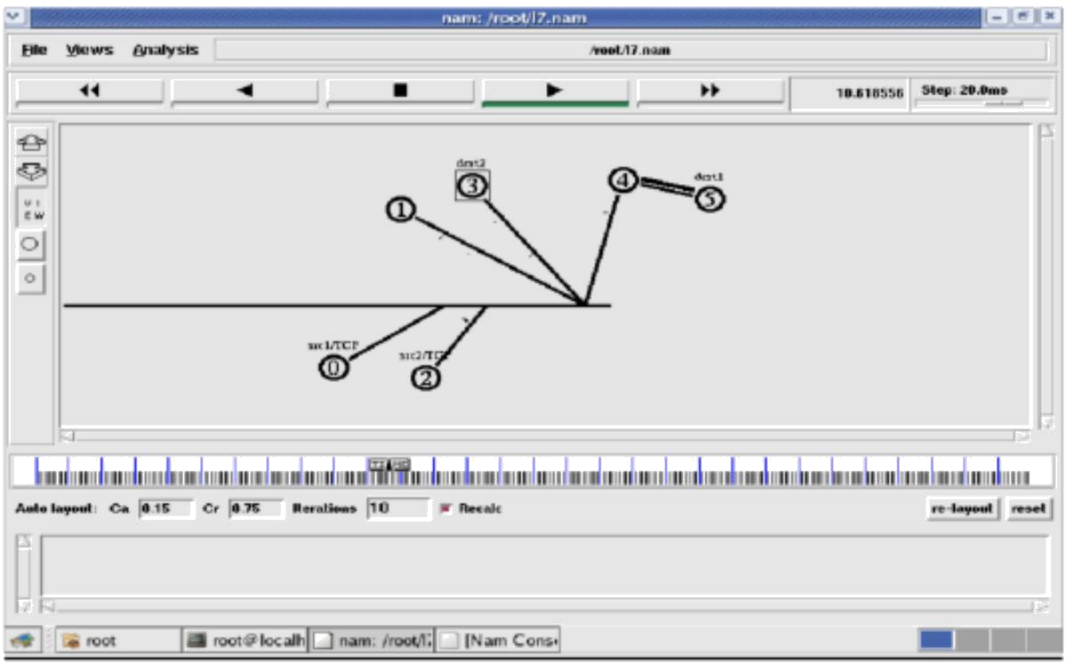
```
BEGIN {
}
{
if($6= ="cwnd_")                    /* don’t leave space after writing cwnd_ */
printf("%f\t%f\t\n",$1,$7);         /* you must put \n in printf */
}
END {
}
```

Steps for Execution

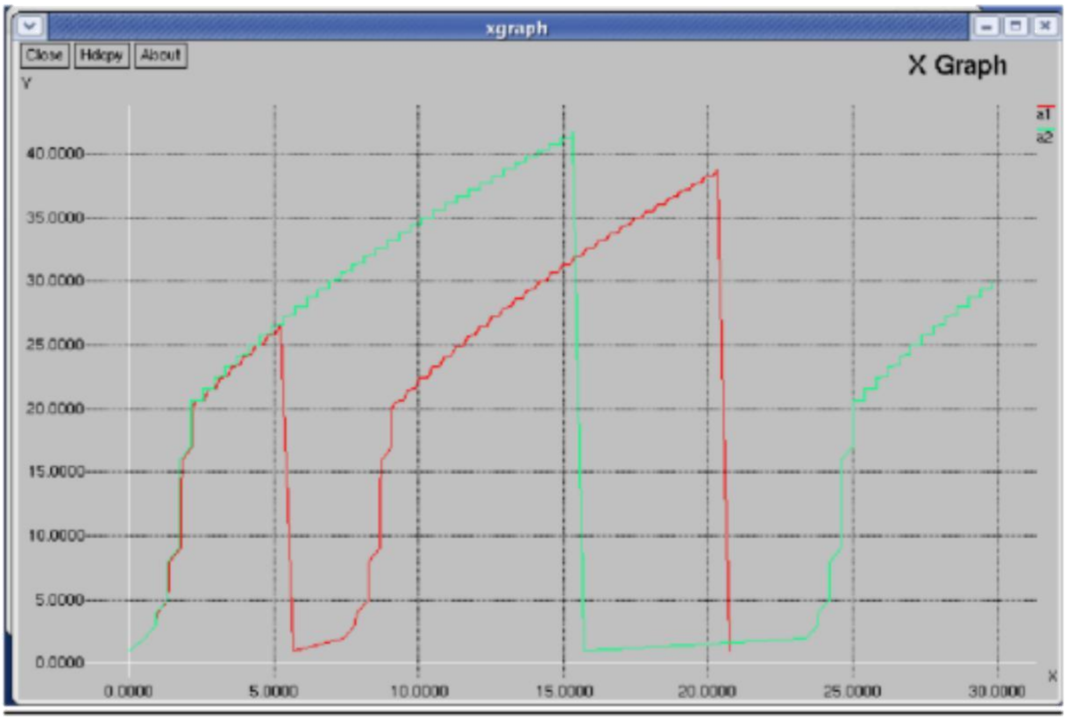
1) Open gedit and type the program. program name should have extension **“.tcl” [root@localhost ~]# gedit lab3.tcl**
2) Save the program by pressing **“CTRL + S”**
3) Open gedit and type awk program. program name should have extension **“.awk” [root@localhost ~]# gedit lab3.awk**
4) Save the program by pressing **“CTRL + S”**
5)Run the simulation program
   [root@localhost~]# ns lab3.tcl
i) Here **“ns”** indicates network simulator. We get the topology shown in the snapshot.
ii) Now press the play button in the simulation window and the simulation will begins.

6) After simulation is completed run awk file to see the output ,
   i. [root@localhost~]# awk –f lab3.awk file1.tr > a1
   ii. [root@localhost~]# awk –f lab3.awk file2.tr > a2
   iii. [root@localhost~]# xgraph a1 a2
7) Here we are using the congestion window trace files i.e. file1.tr and file2.tr and we are redirecting the contents of those files to new files say a1 and a2 using output redirection operator (>).
8) To see the trace file contents open the file as ,
   [root@localhost~]# **vi lab3.tr**

Topology



Output



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

```
set ns [new Simulator]

set tf [open lab4.tr w]
$ns trace-all $tf

set topo [new Topography]
$topo load_flatgrid 1000 1000

set nf [open lab4.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 lab4.nam &
close $tf
exit 0
}
 $ns at 250 "finish"
$ns run
```

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)));
}
```

Steps for execution

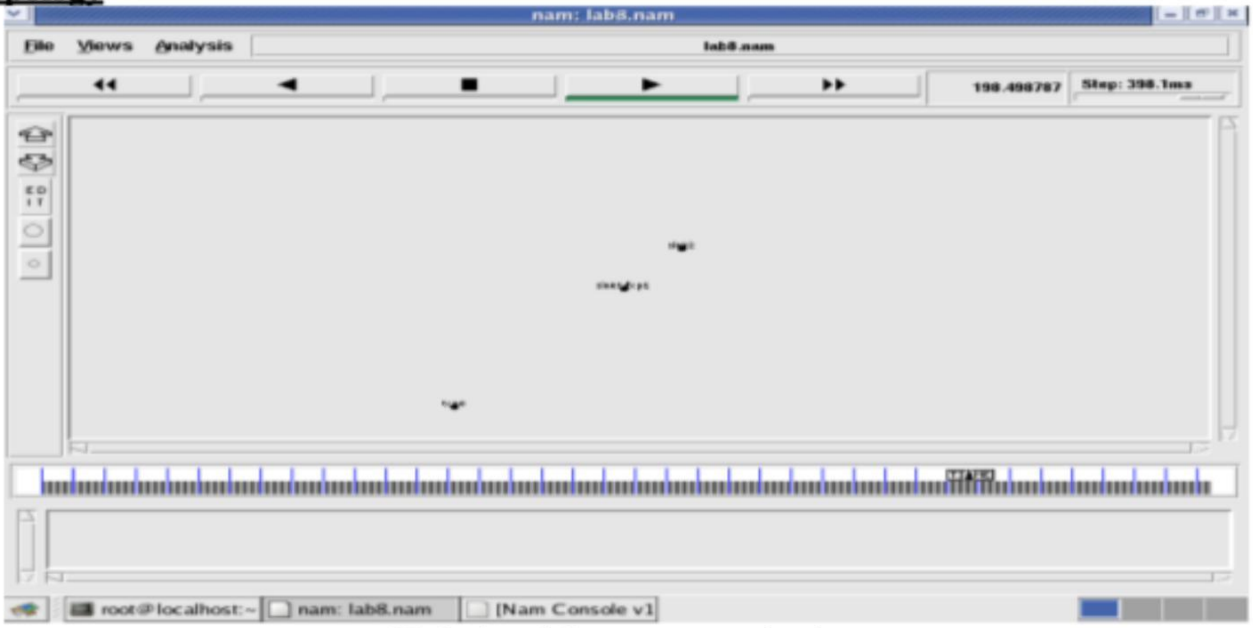
1) Open gedit and type the program. program name should have extension **“.tcl” [root@localhost ~]# gedit lab4.tcl**
2) Save the program by pressing **“CTRL + S”**
3) Open gedit and type awk program. program name should have extension **“.awk” [root@localhost ~]# gedit lab4.awk**
4) Save the program by pressing **“CTRL + S”**
5)Run the simulation program
[root@localhost~]# ns lab4.tcl

i) Here **“ns”** indicates network simulator. We get the topology shown in the snapshot.

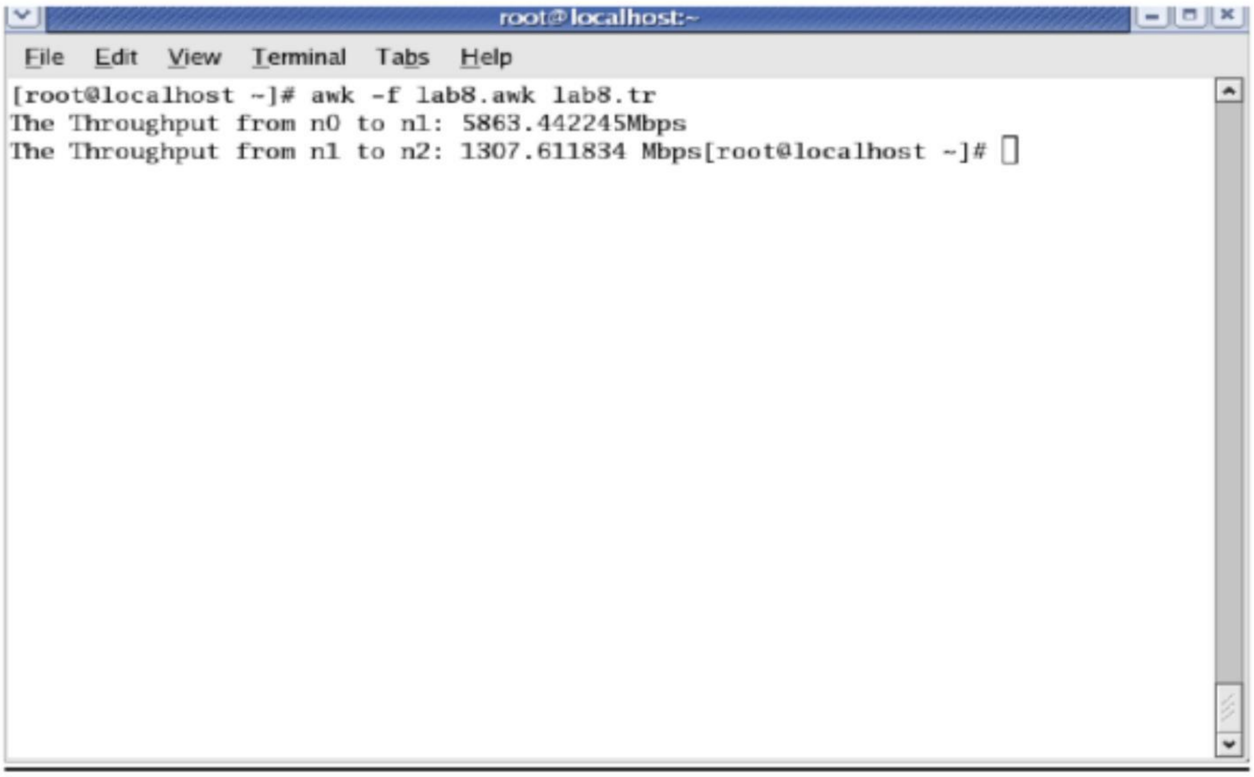
ii) Now press the play button in the simulation window and the simulation will begins.

6) After simulation is completed run awk file to see the
 output , [root@localhost~]# **awk –f lab4.awk lab4.tr**
7) To see the trace file contents open the
 file as , [root@localhost~]# **vi lab4.tr**

Topology



Node 1 and 2 are communicating

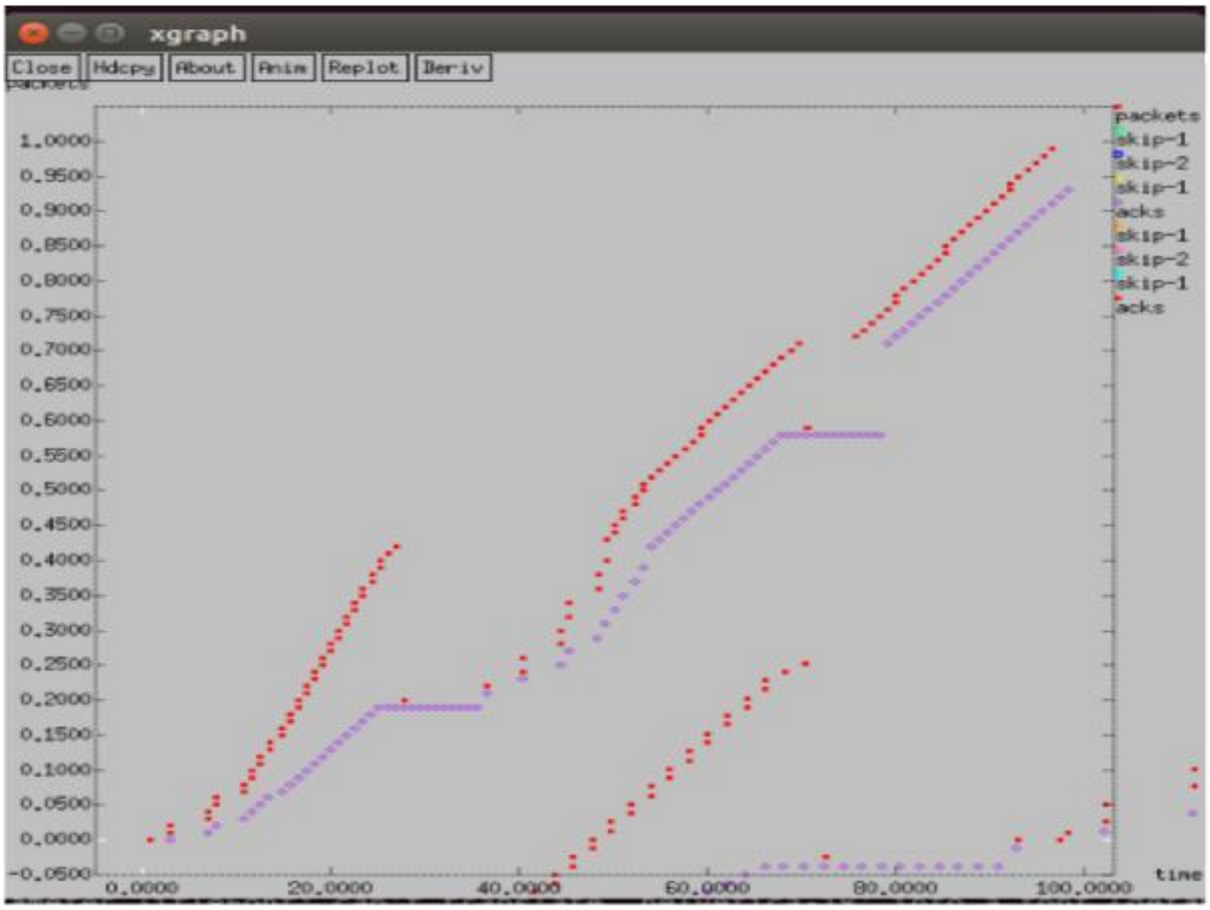


```
[root@localhost ~]# awk -f lab8.awk lab8.tr
The Throughput from n0 to n1: 5863.442245Mbps
The Throughput from n1 to n2: 1307.611834 Mbps[root@localhost ~]#
```

5. Implement and study the performance of GSM on NS2/NS3 (Using MAC layer) or equivalent environment.

```
set ns [new Simulator]

set tf [open out.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]

source port.tcl

switch $opt(type) {
  gsm  -
  gprs  -
  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"
}

$ns at $opt(stop) "stop"
$ns run
```

Output:

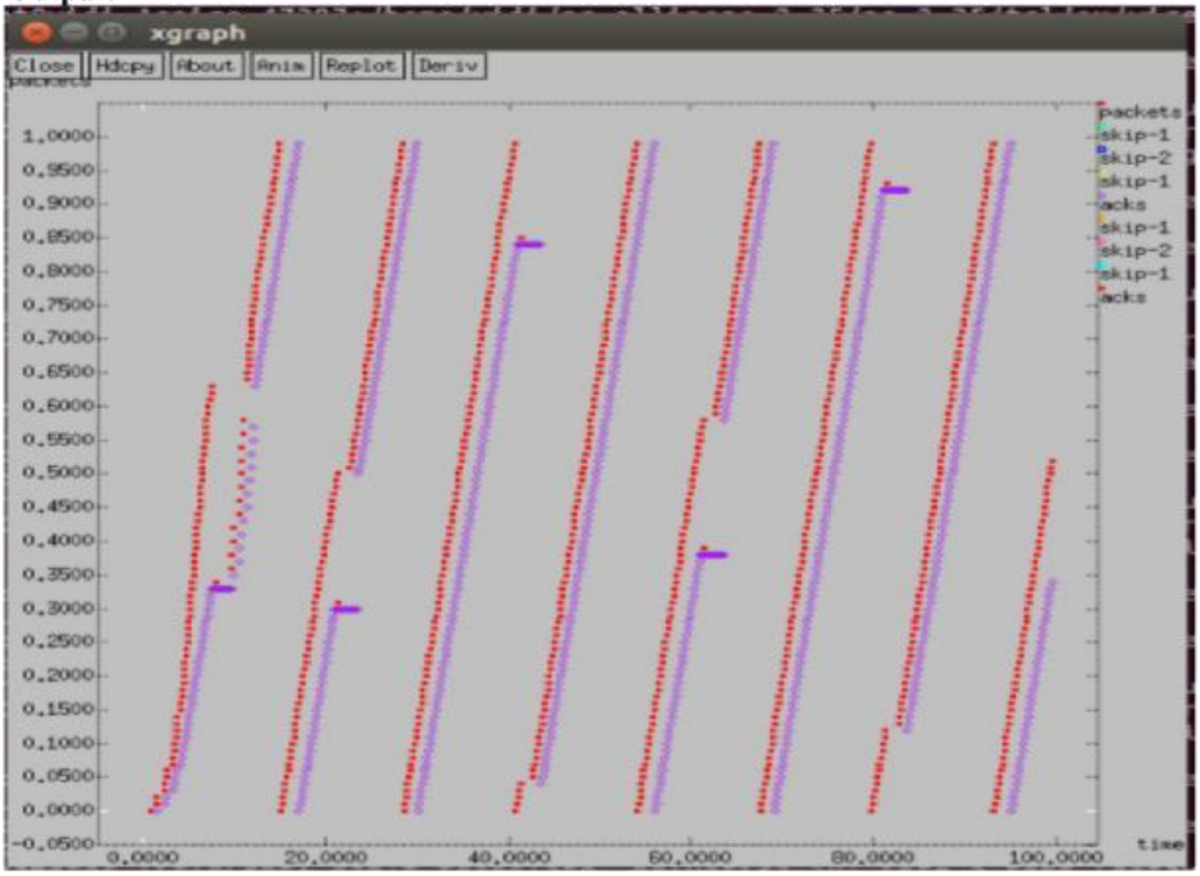


xgraph
Close
Hdcpy
About
Anim
Replot
Deriv
packets
1.0000
0.9500
0.9000
0.8500
0.8000
0.7500
0.7000
0.6500
0.6000
0.5500
0.5000
0.4500
0.4000
0.3500
0.3000
0.2500
0.2000
0.1500
0.1000
0.0500
0.0000
-0.0500
0.0000
20.0000
40.0000
60.0000
80.0000
100.0000
time
packets
skip-1
skip-2
skip-1
acks
skip-1
skip-2
skip-1
acks

6. Implement and study the performance of CDMA on NS2/NS3 (Using stack called Callnet) or equivalent environment.

```
set ns [new Simulator]

set tf [open out.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]

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"
}


$ns at $opt(stop) "stop"
$ns run
```

Output:


xgraph
Close
Hdcpy
About
Anim
Replot
Deriv
packets
1.0000
0.9500
0.9000
0.8500
0.8000
0.7500
0.7000
0.6500
0.6000
0.5500
0.5000
0.4500
0.4000
0.3500
0.3000
0.2500
0.2000
0.1500
0.1000
0.0500
0.0000
-0.0500
0.0000
20.0000
40.0000
60.0000
80.0000
100.0000
time
packets
skip-1
skip-2
skip-1
acks
skip-1
skip-2
skip-1
acks