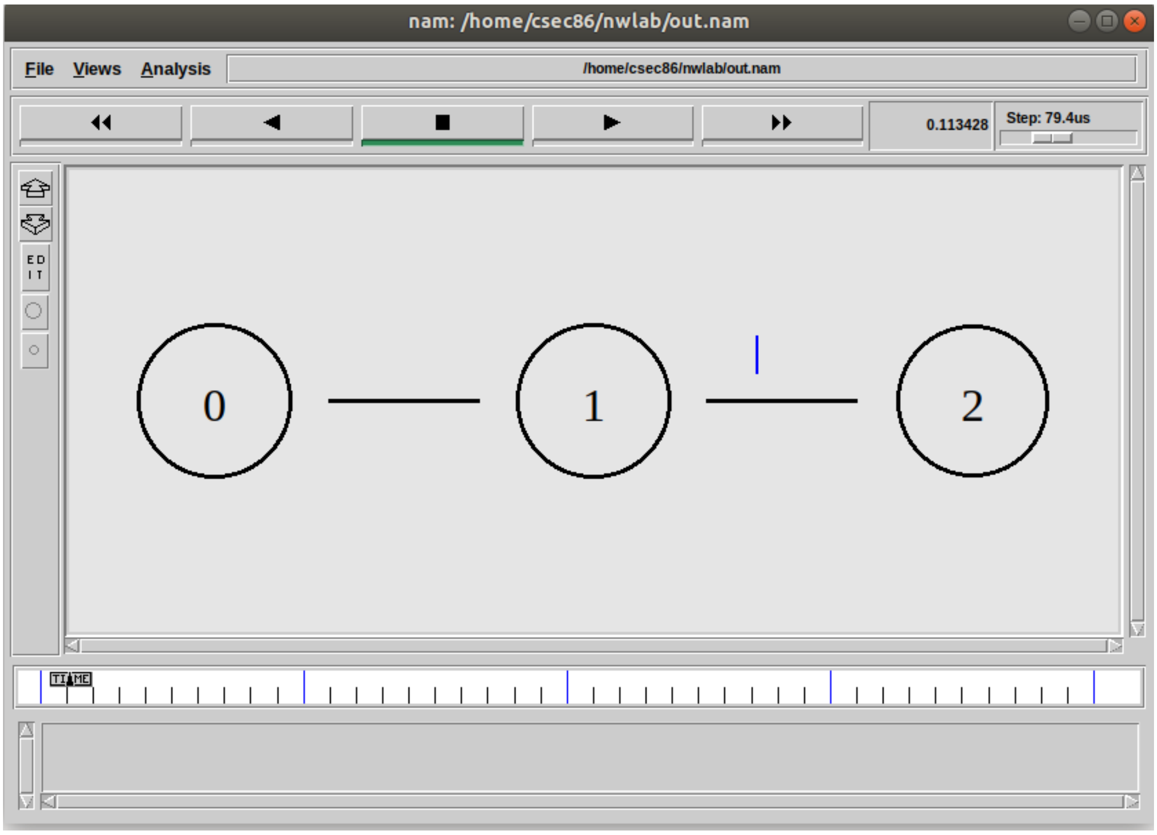
PROGRAM CODE:

```
#Create a simulator object
set ns [new Simulator]
#Open the nam trace file
set nf [open outt.nam w]
$ns namtrace-all $nf
$ns color 1 Blue
#Define a 'finish' procedure
proc finish {} {
global ns nf
$ns flush-trace
#Close the trace file
close $nf
#Execute nam on the trace file
exec nam outt.nam &
exit 0
}
# Creating Nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]

#Setting Links
$ns duplex-link $n0 $n1 10Mb 10ms DropTail
$ns duplex-link $n1 $n2 2Mb 10ms DropTail

#Setting Topology
$ns duplex-link-op $n0 $n1 orient right
$ns duplex-link-op $n1 $n2 orient right

#Setting Queue Limit
$ns queue-limit $n0 $n1 8
$ns queue-limit $n1 $n2 8
#Setup a TCP connection over 0 and 2 and its flow id, window size, packet size
set tcp [new Agent/TCP]
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $n2 $sink
$ns connect $tcp $sink
$tcp set fid_ 1
$tcp set window_ 16
$tcp set packetSize_ 552
#Setup a FTP over TCP connection
set ftp [new Application/FTP]
$ftp attach-agent $tcp
```

```
$ftp set type_ FTP

#Start and stop  ftp
$ns at 0.1 "$ftp start"
$ns at 4.0 "$ftp stop"

#Call the finish procedure after 5 seconds of simulation time
$ns at 5.0 "finish"
#Run the simulation
$ns run
```

OUTPUT:

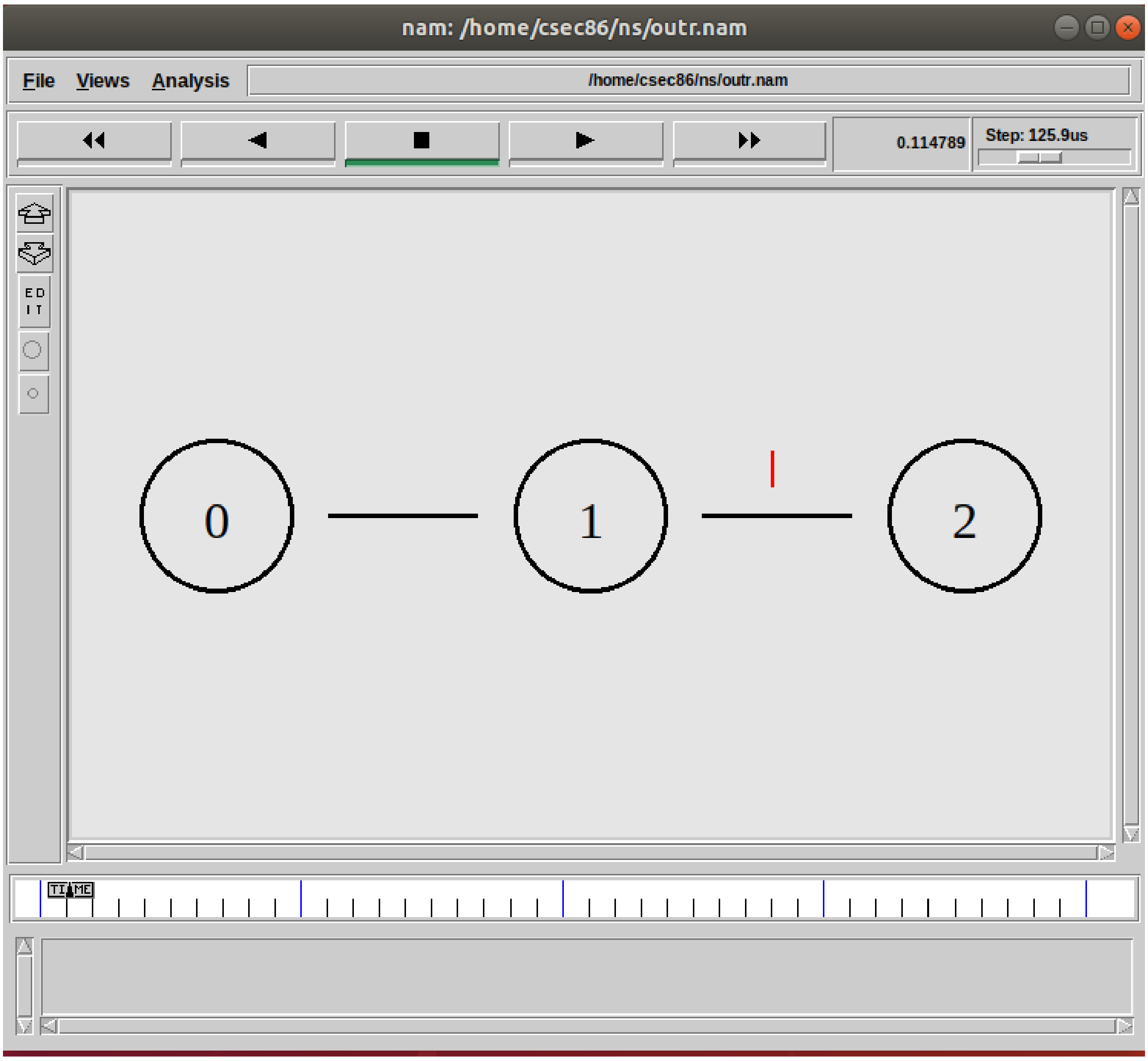


PROGRAM CODE:

```
#Create a simulator object
set ns [new Simulator]
#Open the nam trace file
set nf [open outr.nam w]
$ns namtrace-all $nf
$ns color 1 Red
#Define a 'finish' procedure
proc finish {} {
global ns nf
$ns flush-trace
#Close the trace file
close $nf
#Execute nam on the trace file
exec nam outr.nam &
exit 0
}
# Creating Nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]

#Setting Links
$ns duplex-link $n0 $n1 10Mb 10ms DropTail
$ns duplex-link $n1 $n2 2Mb 10ms DropTail

#Setting Topology
$ns duplex-link-op $n0 $n1 orient right
$ns duplex-link-op $n1 $n2 orient right

#Setting Queue Limit
$ns queue-limit $n0 $n1 8
$ns queue-limit $n1 $n2 8
#Setup a TCP connection over 0 and 2 and its flow id, window size, packet size
set tcp [new Agent/TCP/Reno]
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $n2 $sink
$ns connect $tcp $sink
$tcp set fid_ 1
$tcp set window_ 16
$tcp set packetSize_ 552
#Setup a FTP over TCP connection
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ftp set type_ FTP
```

```
#Start and stop  ftp
$ns at 0.1 "$ftp start"
$ns at 4.0 "$ftp stop"

#Call the finish procedure after 5 seconds of simulation time
$ns at 5.0 "finish"
#Run the simulation
$ns run
```

OUTPUT:



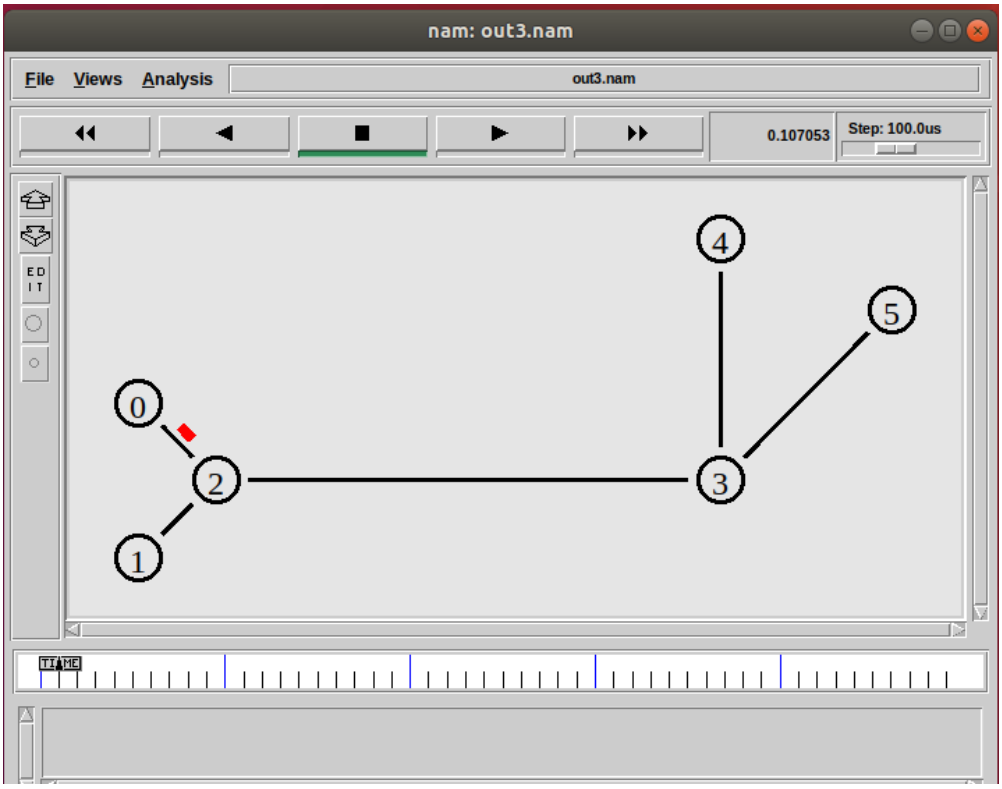
PROGRAM CODE:

```
#Create a simulator object
set ns [new Simulator]
```

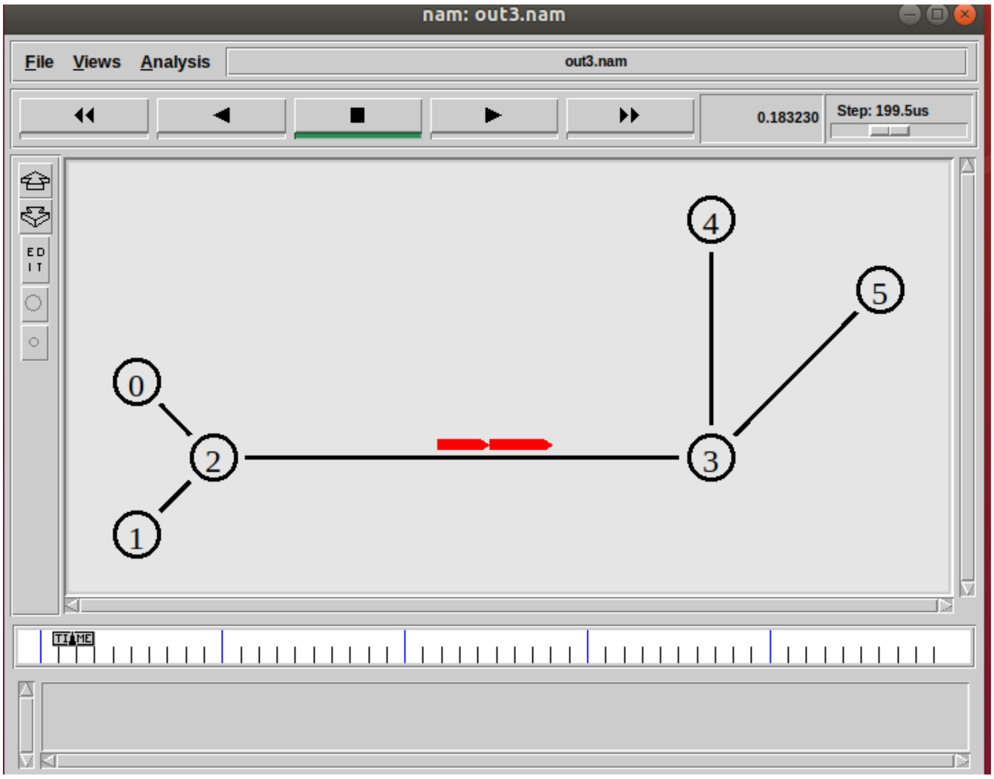
```
#Open the nam trace file
set nf [open out3.nam w]
$ns namtrace-all $nf
$ns color 1 Blue
$ns color 2 Red
#Define a 'finish' procedure
proc finish {} {
global ns nf
$ns flush-trace
#Close the trace file
close $nf
#Execute nam on the trace file
exec nam out3.nam &
exit 0
}
# Creating Nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
#Setting Links
$ns duplex-link $n0 $n2 2Mb 10ms DropTail
$ns duplex-link $n1 $n2 2Mb 10ms DropTail
$ns simplex-link $n2 $n3 0.3Mb 100ms DropTail
$ns simplex-link $n3 $n2 0.3Mb 100ms DropTail
$ns duplex-link $n3 $n4 0.5Mb 40ms DropTail
$ns duplex-link $n3 $n5 0.5Mb 40ms DropTail
#Setting Topology
$ns duplex-link-op $n0 $n2 orient right-down
$ns duplex-link-op $n1 $n2 orient right-up
$ns duplex-link-op $n2 $n3 orient left-up
$ns duplex-link-op $n3 $n2 orient right
$ns duplex-link-op $n3 $n4 orient up
$ns duplex-link-op $n3 $n5 orient right-up
#Setting Queue Limit
$ns queue-limit $n2 $n3 10
#Setup a TCP connection over 0 and 4 and its flow id, window size, packet size
set tcp [new Agent/TCP/Newreno]
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSink/DelAck]
$ns attach-agent $n4 $sink
$ns connect $tcp $sink
$tcp set fid_ 1
$tcp set window_ 8000
```

```
$tcp set packetSize_ 552
#Setup a FTP over TCP connection
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ftp set type_ FTP
#Create a UDP agent and attach it to node n0
set udp [new Agent/UDP]
$ns attach-agent $n0 $udp
# Create a CBR traffic source and attach it to udp0
set cbr [new Application/Traffic/CBR]
$cbr set type_ CBR
$cbr set packet_size_ 1000
$cbr set rate_ 0.01mb
$cbr set random_ false
$cbr attach-agent $udp
#Create a Null agent (a traffic sink) and attach it to node n1
set null [new Agent/Null]
$ns attach-agent $n5 $null
#Connect the traffic source with the traffic sink
$ns connect $udp $null
#Set Flow ID, Packet Size and Window Size
$udp set fid_ 2
$udp set window_ 8000
$udp set packetSize_ 552
#Start and stop the cbr and ftp
$ns at 0.1 "$cbr start"
$ns at 1.0 "$ftp start"
$ns at 4.5 "$ftp stop"
$ns at 5.0 "$cbr stop"
#Call the finish procedure after 5 seconds of simulation time
$ns at 5.0 "finish"
#Run the simulation
$ns run
```

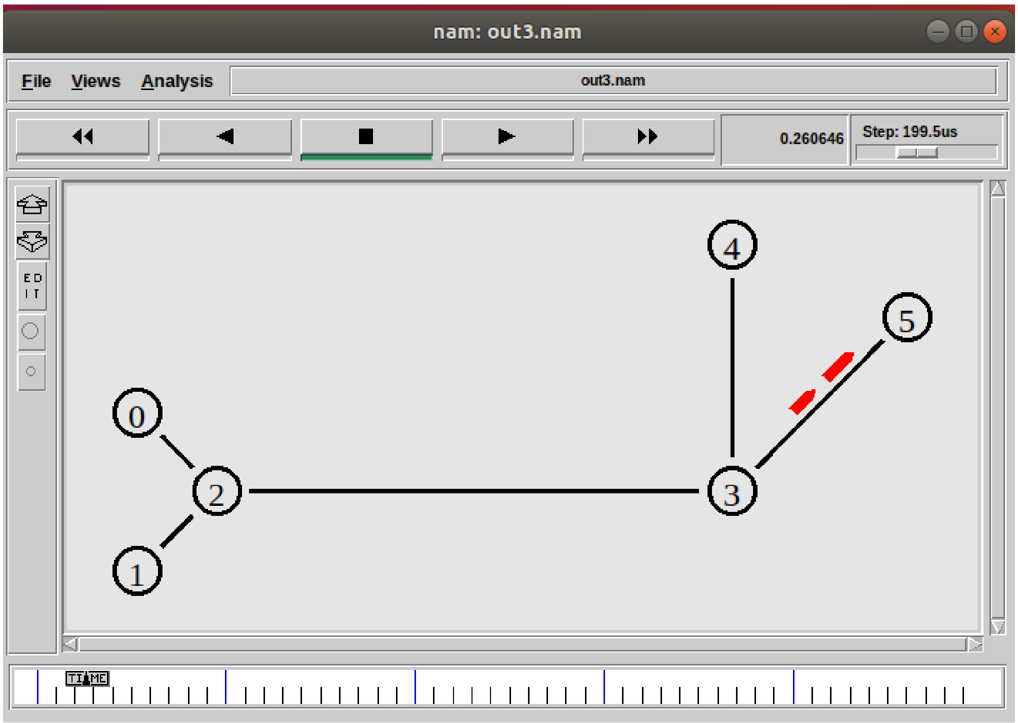
OUTPUT:

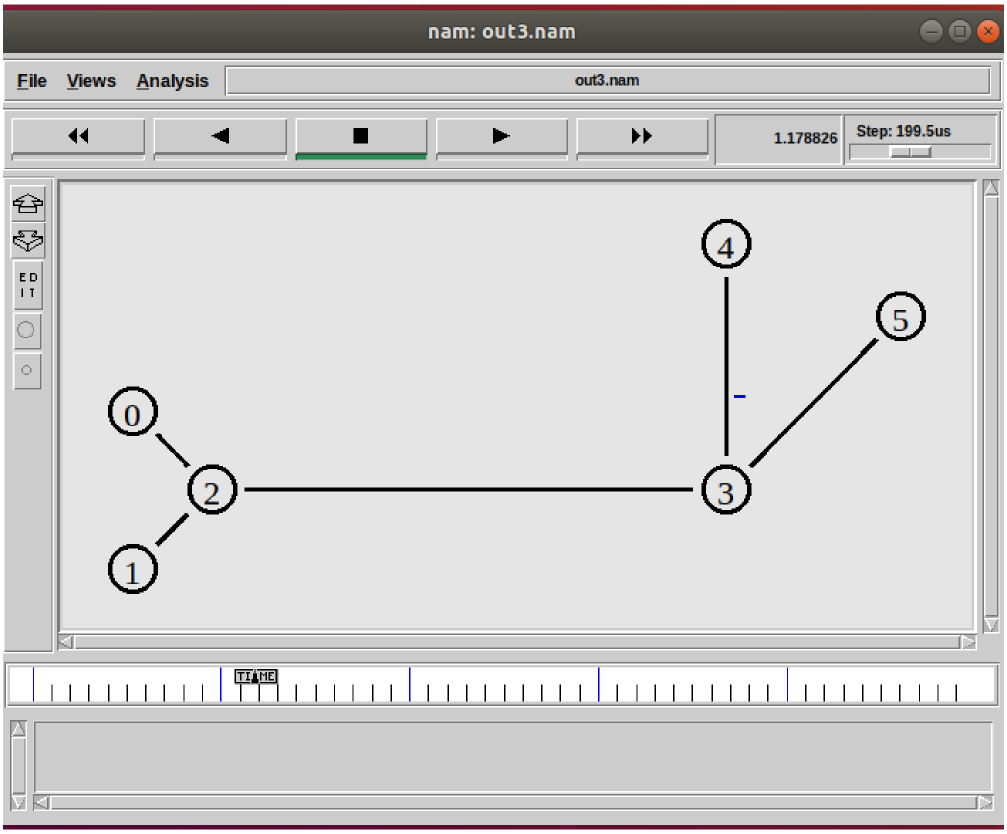


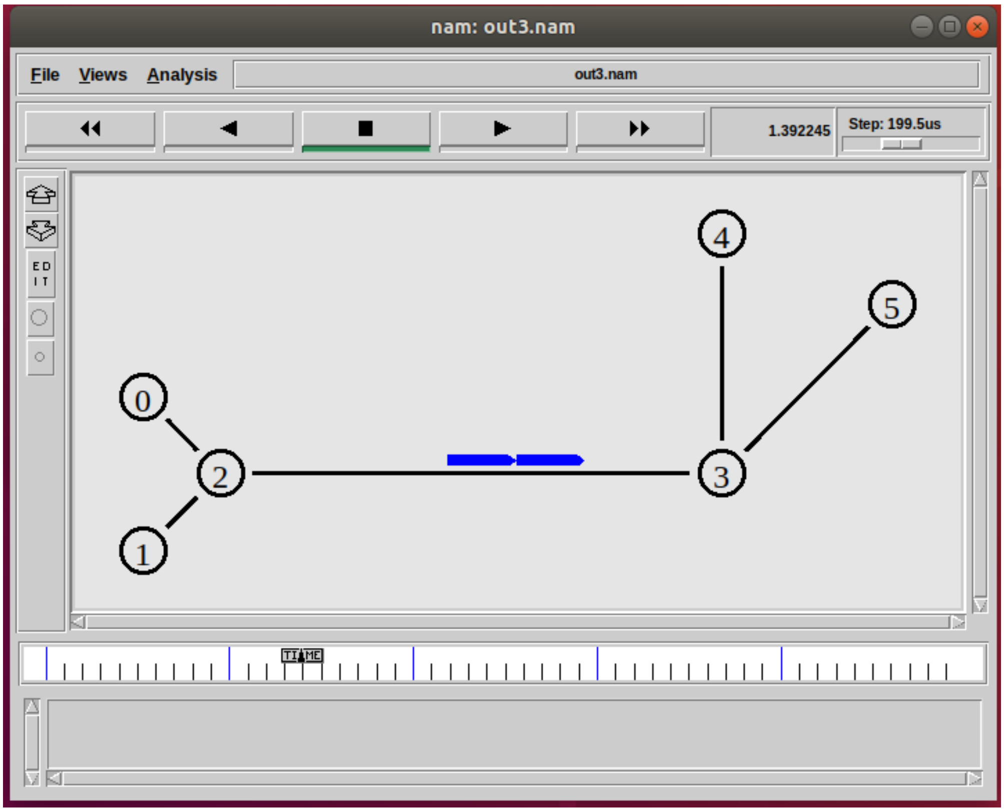
nam: out3.nam
File
Views
Analysis
out3.nam
0.107053
Step: 100.0us
EDIT
TIME
0
1
2
3
4
5

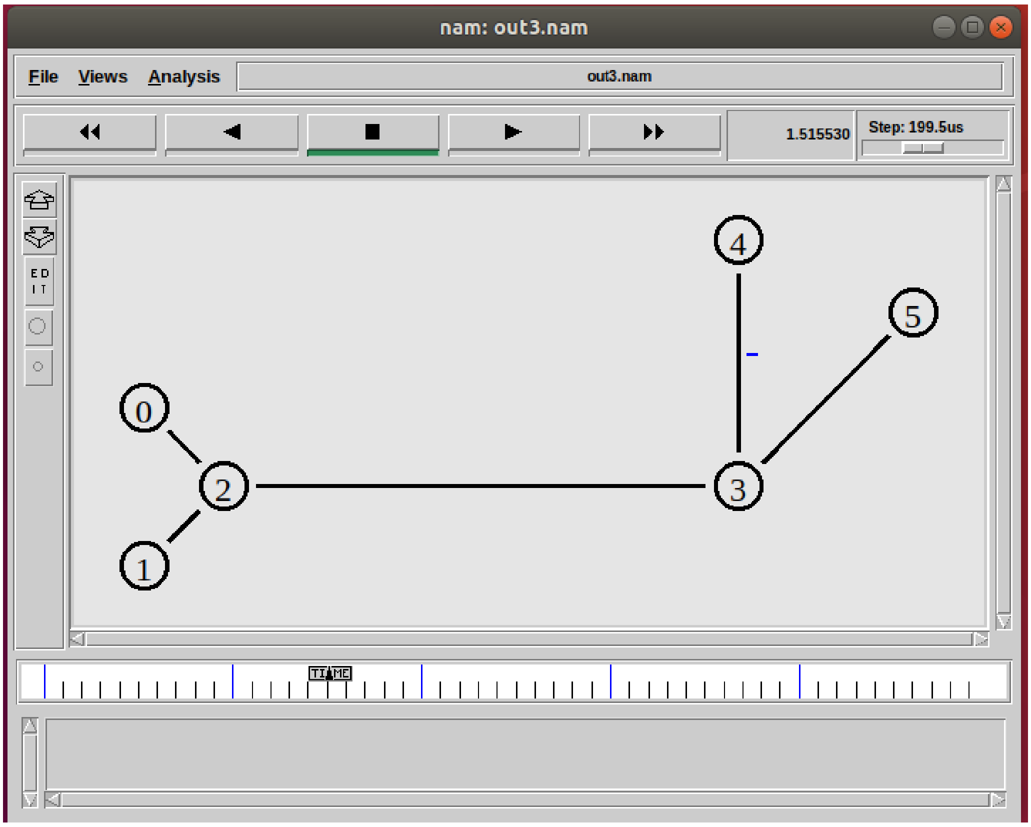


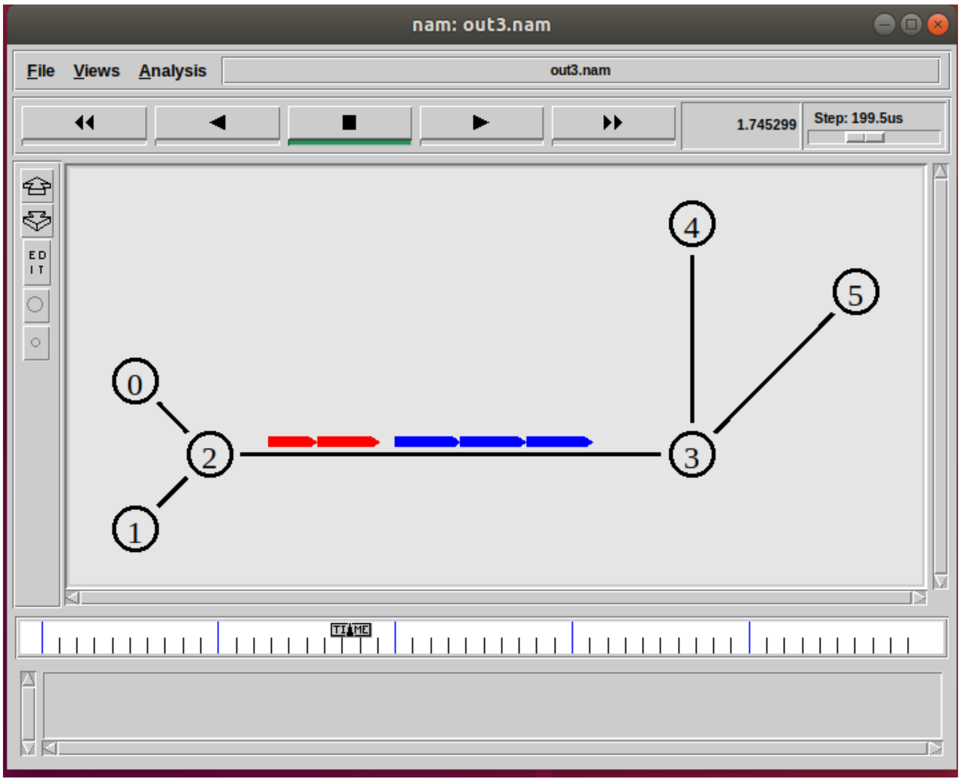
nam: out3.nam
File
Views
Analysis
out3.nam
0.183230
Step: 199.5us
EDIT
TIME
0
1
2
3
4
5

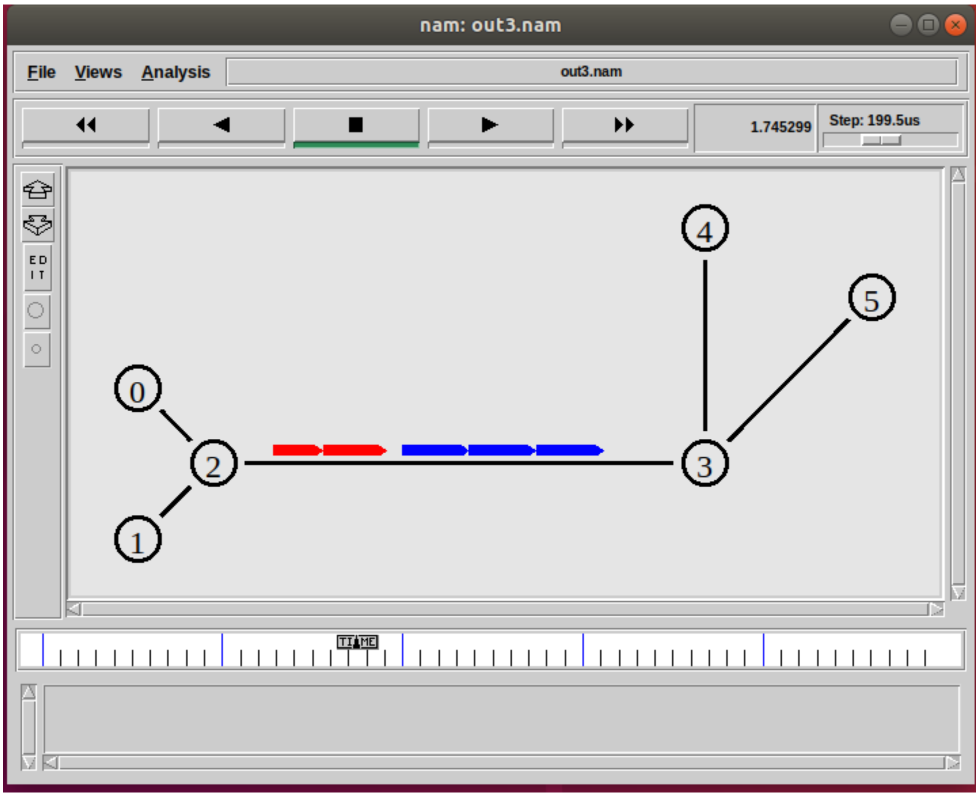

nam: out3.nam
File
Views
Analysis
out3.nam
0.260646
Step: 199.5us
EDIT
TIME
0
1
2
3
4
5

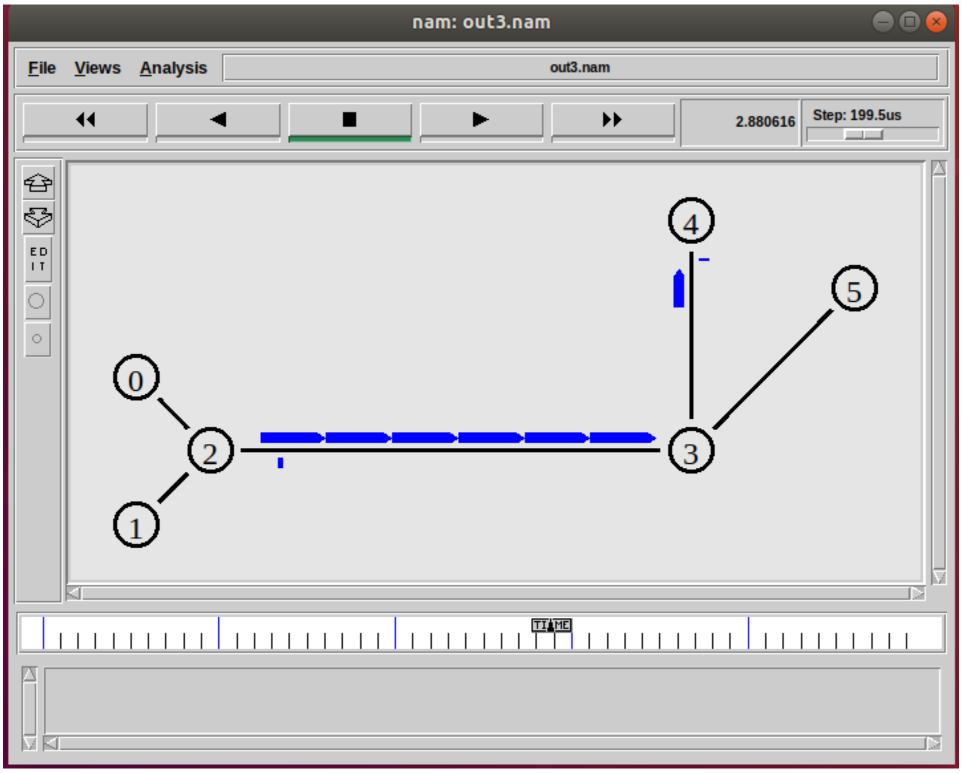

nam: out3.nam
File
Views
Analysis
out3.nam
1.178826
Step: 199.5us
EDIT
TIME
0
1
2
3
4
5

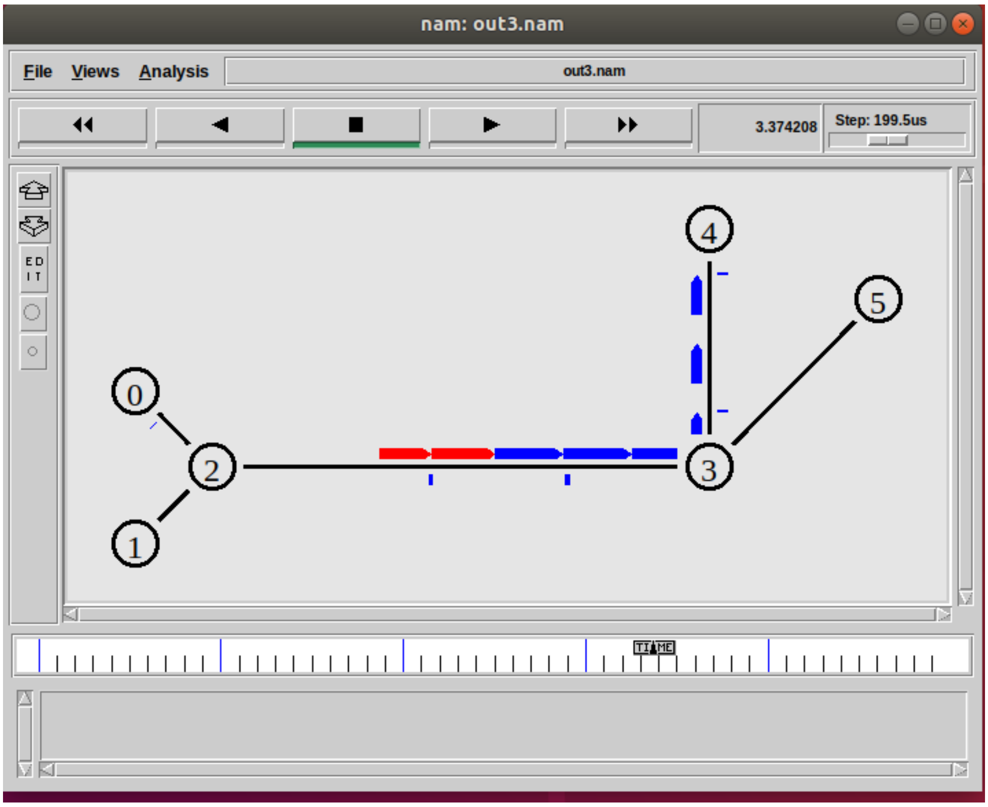

nam: out3.nam
File
Views
Analysis
out3.nam
1.392245
Step: 199.5us
EDIT
TIME
0
1
2
3
4
5

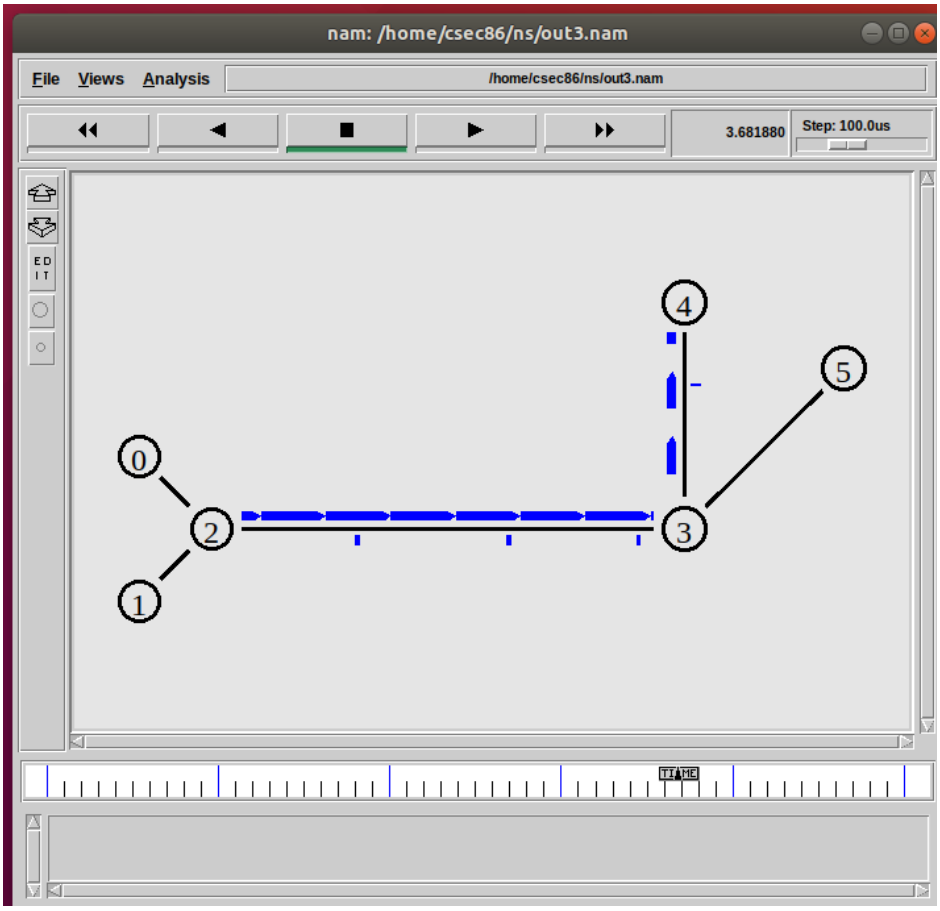

nam: out3.nam
File
Views
Analysis
out3.nam
1.515530
Step: 199.5us
EDIT
TIME
0
1
2
3
4
5

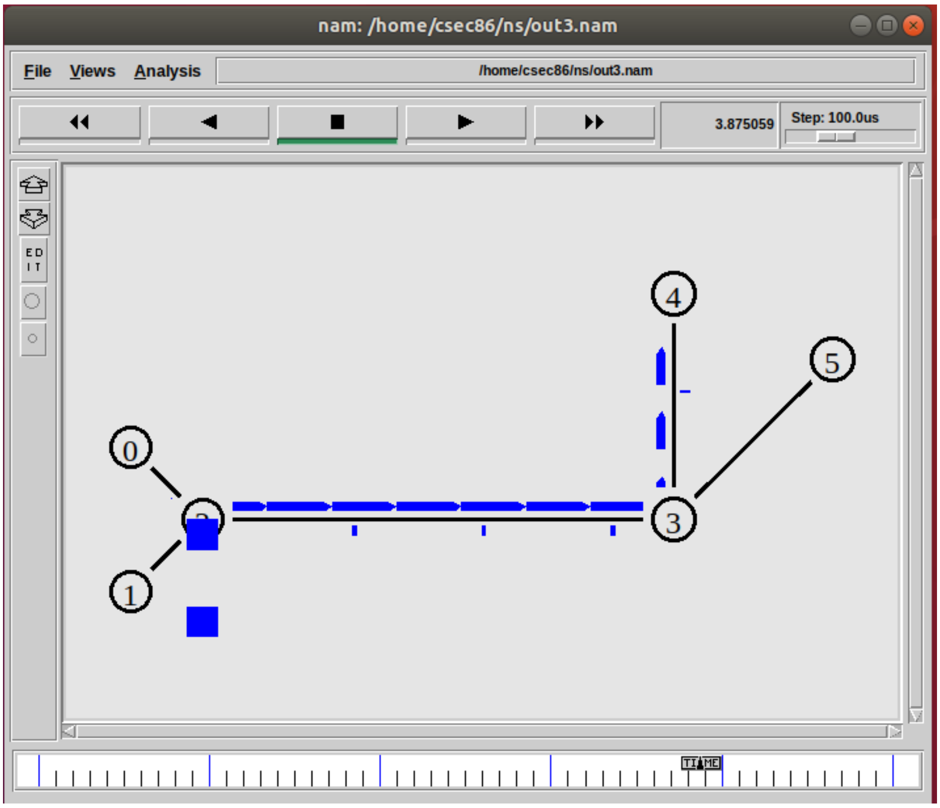

nam: out3.nam
File
Views
Analysis
out3.nam
1.745299
Step: 199.5us
EDIT
TIME
0
1
2
3
4
5

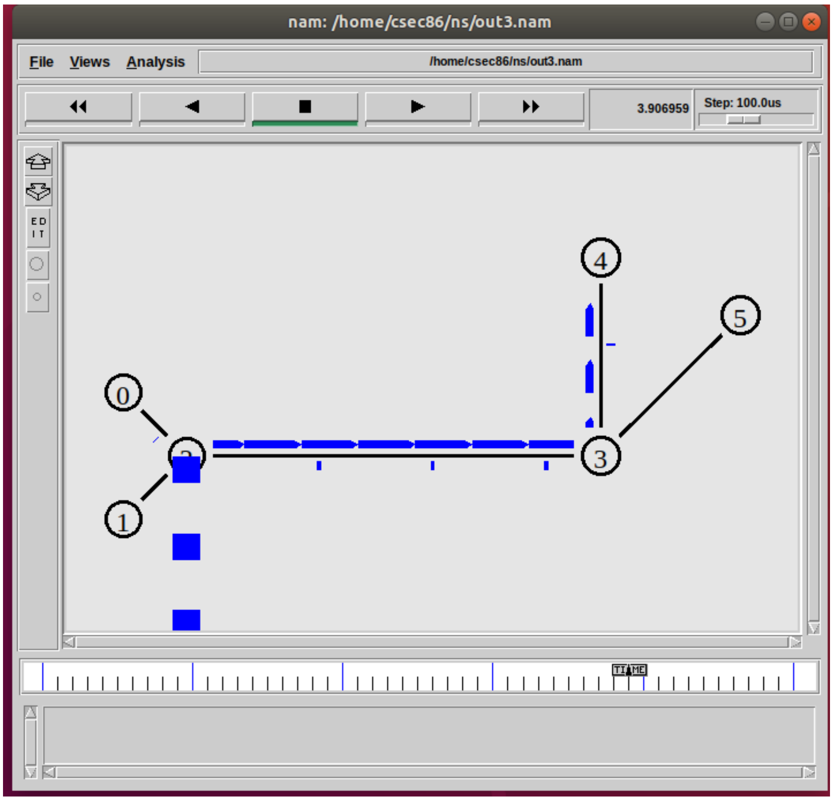

nam: out3.nam
File
Views
Analysis
out3.nam
1.745299
Step: 199.5us
EDIT
TIME
0
1
2
3
4
5

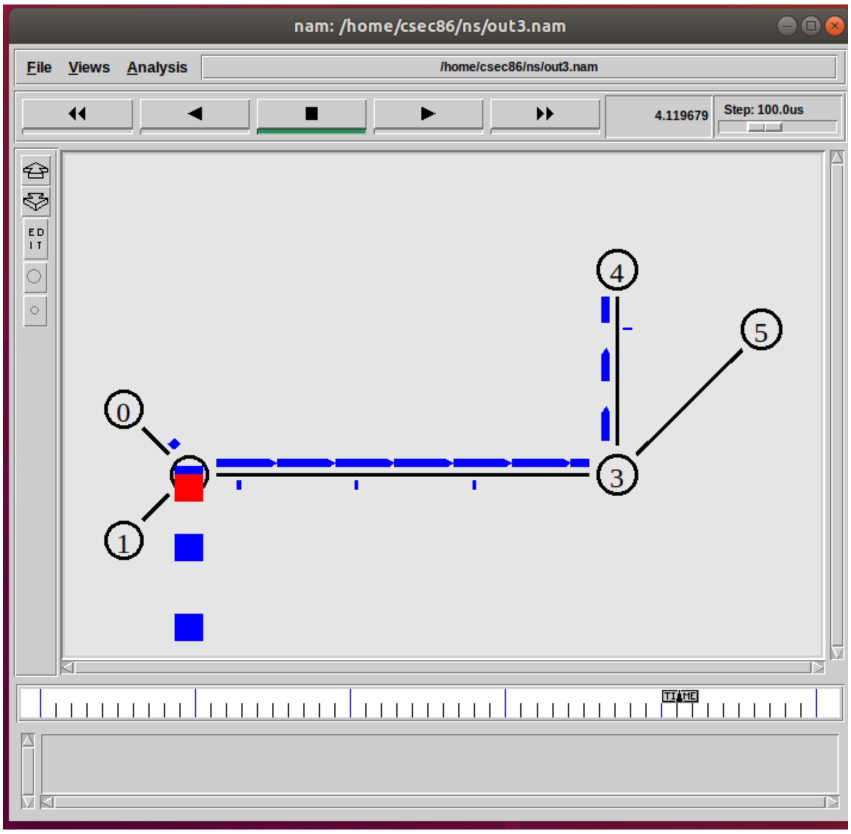

nam: out3.nam
File
Views
Analysis
out3.nam
2.880616
Step: 199.5us
EDIT
TIME
0
1
2
3
4
5

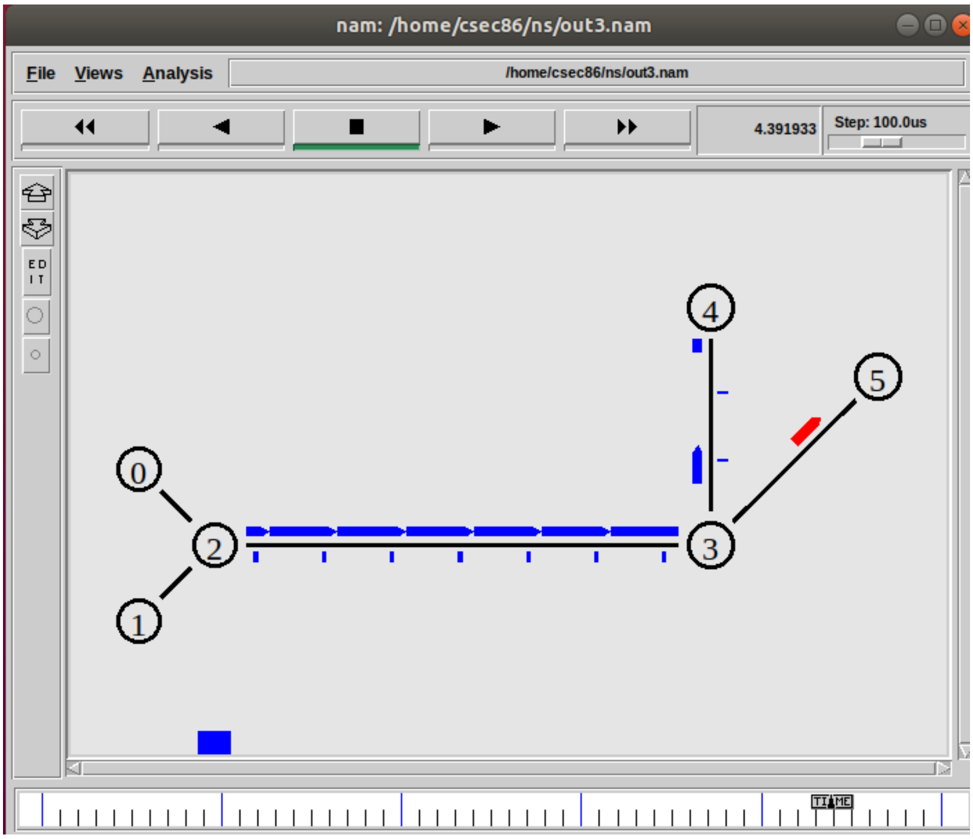

nam: out3.nam
File
Views
Analysis
out3.nam
3.374208
Step: 199.5us
EDIT
TIME
0
1
2
3
4
5

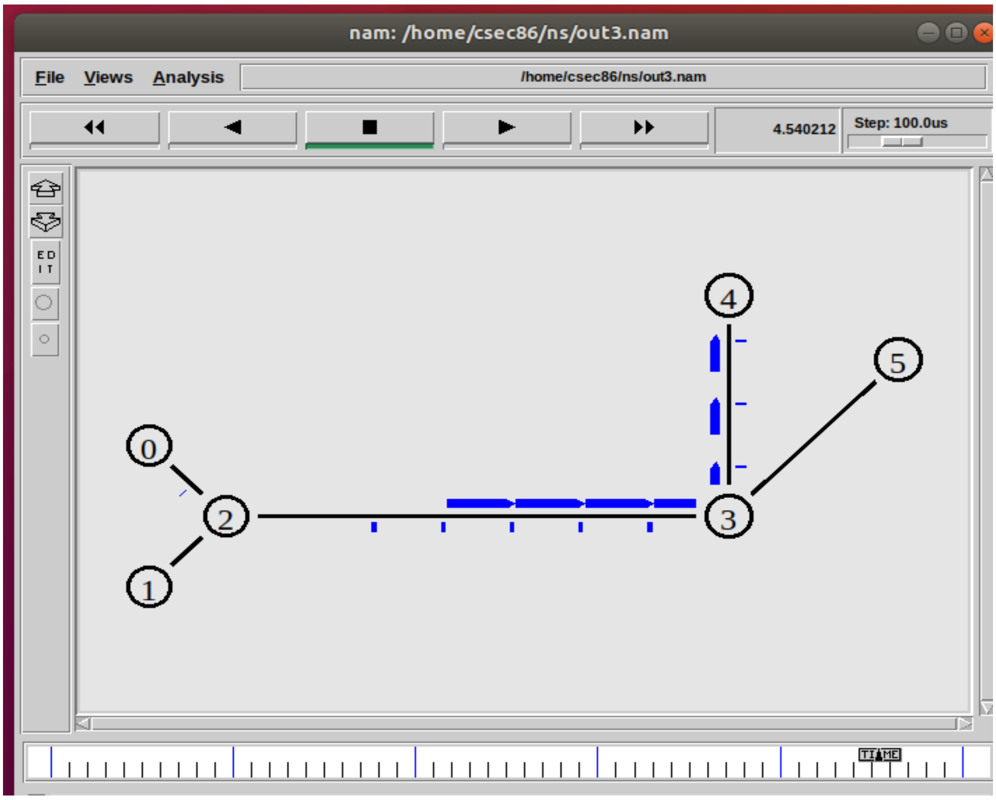

nam: /home/csec86/ns/out3.nam
File
Views
Analysis
/home/csec86/ns/out3.nam
3.681880
Step: 100.0us
EDIT
TIME
0
1
2
3
4
5


nam: /home/csec86/ns/out3.nam
File
Views
Analysis
/home/csec86/ns/out3.nam
3.875059
Step: 100.0us
EDIT
TIME
0
1
2
3
4
5


nam: /home/csec86/ns/out3.nam
File
Views
Analysis
/home/csec86/ns/out3.nam
3.906959
Step: 100.0us
EDIT
0
1
2
3
4
5
TIME


nam: /home/csec86/ns/out3.nam
File
Views
Analysis
/home/csec86/ns/out3.nam
4.119679
Step: 100.0us
EDIT
0
1
2
3
4
5
TIME


nam: /home/csec86/ns/out3.nam
File
Views
Analysis
/home/csec86/ns/out3.nam
4.391933
Step: 100.0us
EDIT
0
1
2
3
4
5
TIME

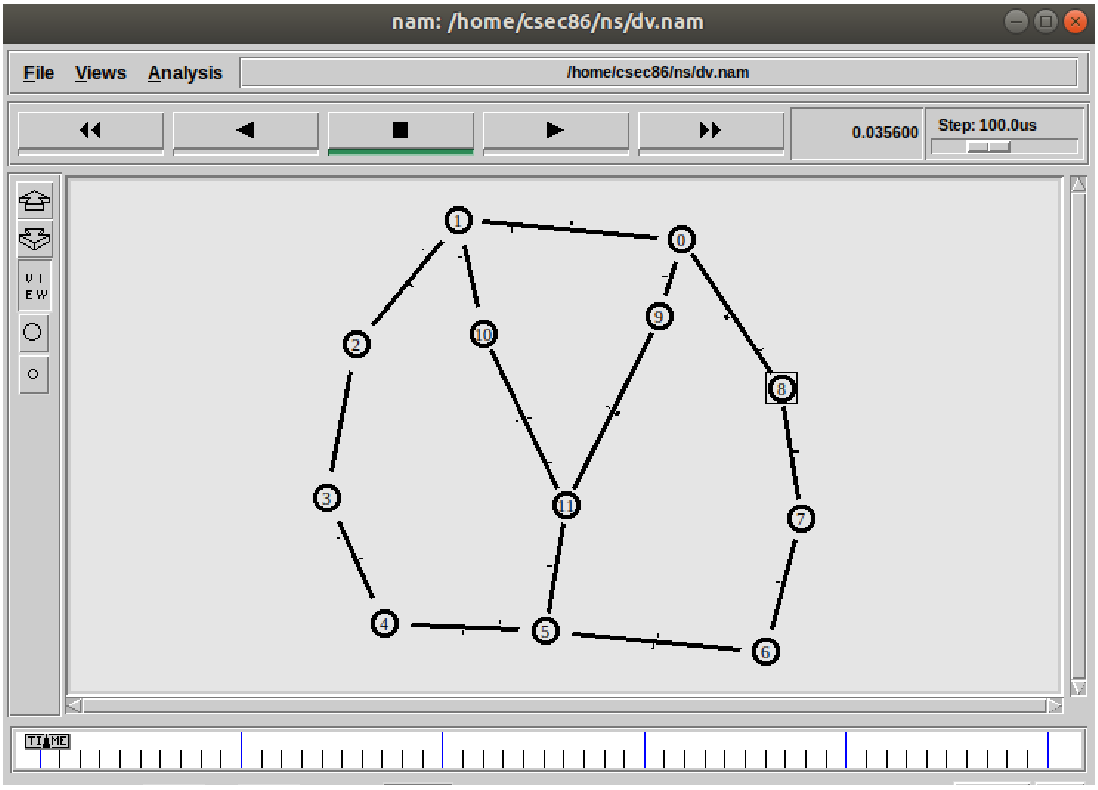

nam: /home/csec86/ns/out3.nam
File
Views
Analysis
/home/csec86/ns/out3.nam
4.540212
Step: 100.0us
EDIT
0
1
2
3
4
5
TIME

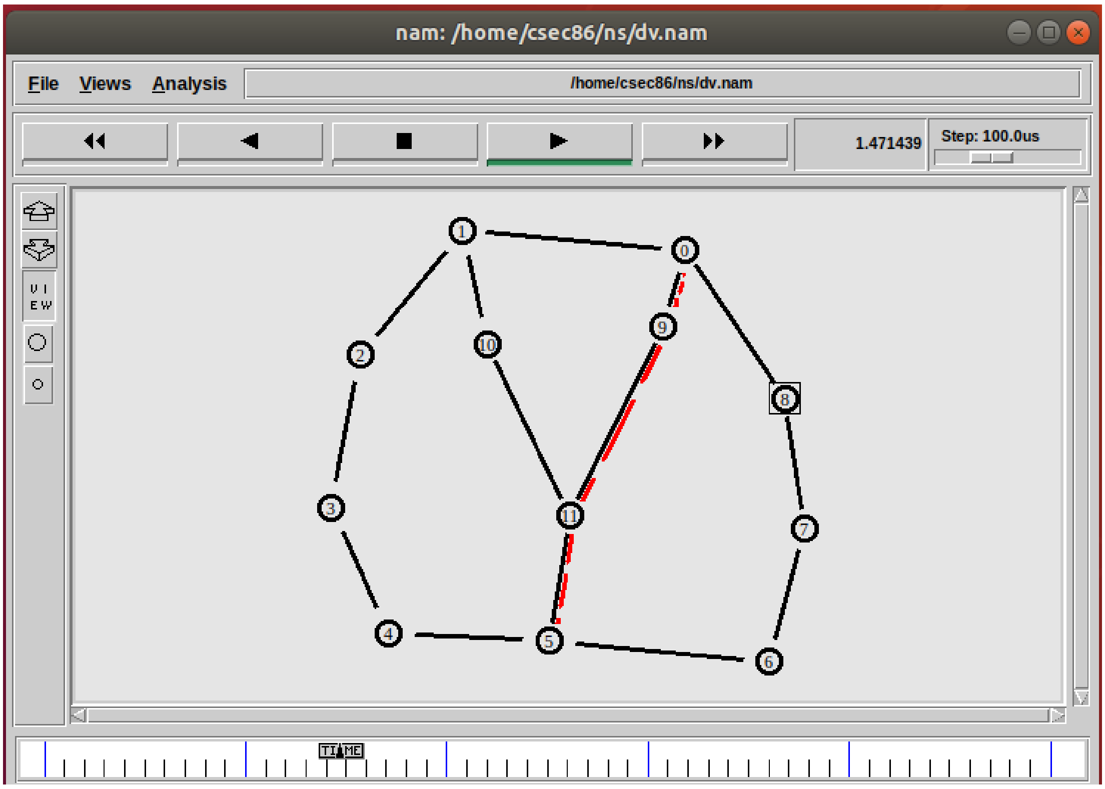
PROGRAM CODE:

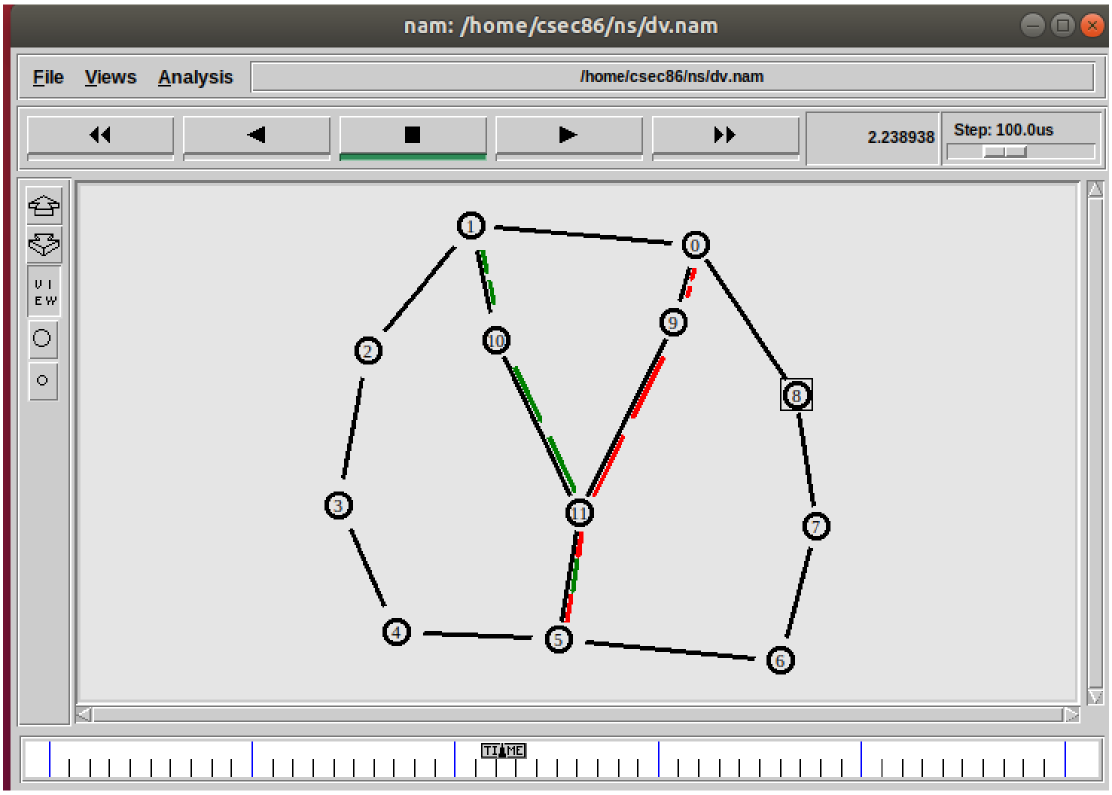
```
set ns [new Simulator]
set nr [open dv.tr w]
$ns trace-all $nr
set nf [open dv.nam w]
$ns namtrace-all $nf
proc finish { } {
global ns nr nf
$ns flush-trace
close $nf
close $nr
exec nam dv.nam &
exit 0
}

for { set i 0 } { $i < 12} { incr i 1 } {
set n($i) [$ns node]}

for {set i 0} {$i < 8} {incr i} {
$ns duplex-link $n($i) $n([expr $i+1]) 1Mb 10ms DropTail }
$ns duplex-link $n(0) $n(8) 1Mb 10ms DropTail
$ns duplex-link $n(1) $n(10) 1Mb 10ms DropTail
$ns duplex-link $n(0) $n(9) 1Mb 10ms DropTail
$ns duplex-link $n(9) $n(11) 1Mb 10ms DropTail
$ns duplex-link $n(10) $n(11) 1Mb 10ms DropTail
$ns duplex-link $n(11) $n(5) 1Mb 10ms DropTail

set udp0 [new Agent/UDP]
$ns attach-agent $n(0) $udp0
set cbr0 [new Application/Traffic/CBR]
$cbr0 set packetSize_ 500
$cbr0 set interval_ 0.005
$cbr0 attach-agent $udp0
set null0 [new Agent/Null]
$ns attach-agent $n(5) $null0
$ns connect $udp0 $null0

set udp1 [new Agent/UDP]
$ns attach-agent $n(1) $udp1
set cbr1 [new Application/Traffic/CBR]
$cbr1 set packetSize_ 500
$cbr1 set interval_ 0.005
$cbr1 attach-agent $udp1
set null0 [new Agent/Null]
$ns attach-agent $n(5) $null0
```

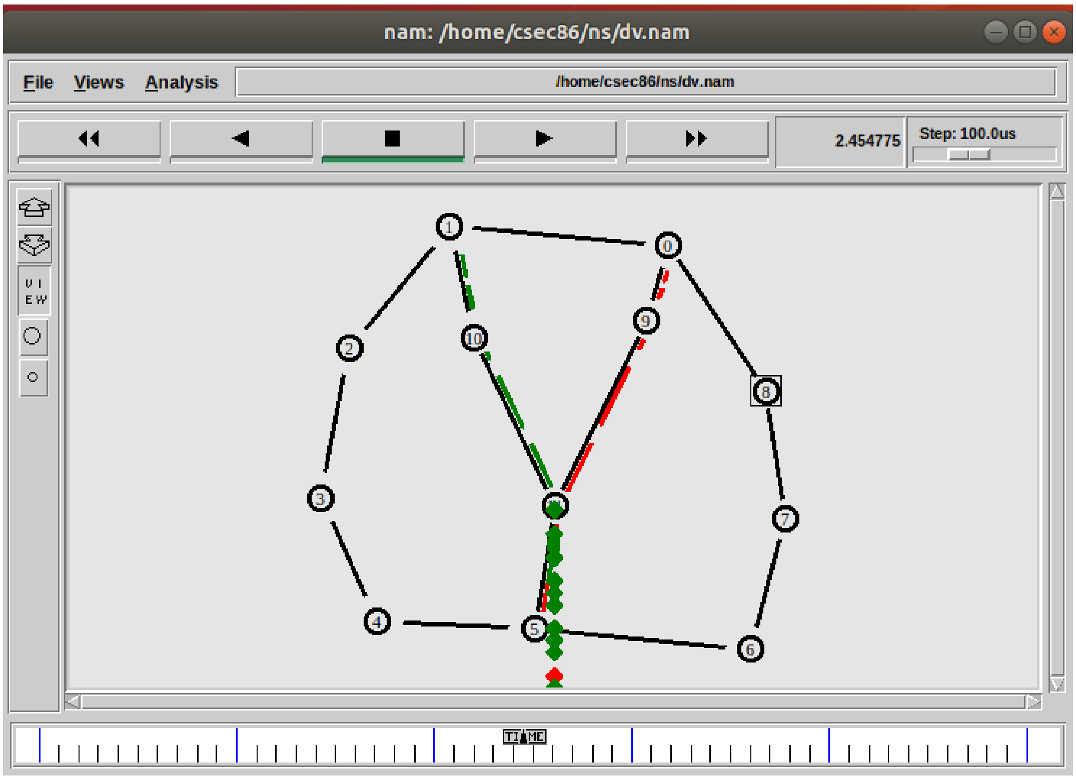
```
$ns connect $udp1 $null0

$ns rtproto DV

$ns rtmodel-at 3.0 down $n(11) $n(5)
$ns rtmodel-at 3.0 down $n(7) $n(6)
$ns rtmodel-at 4.0 up $n(11) $n(5)
$ns rtmodel-at 4.0 up $n(7) $n(6)

$udp0 set fid_ 1
$udp1 set fid_ 2

$ns color 1 Red
$ns color 2 Green

$ns at 1.0 "$cbr0 start"
$ns at 2.0 "$cbr1 start"

$ns at 5 "finish"
$ns run
```

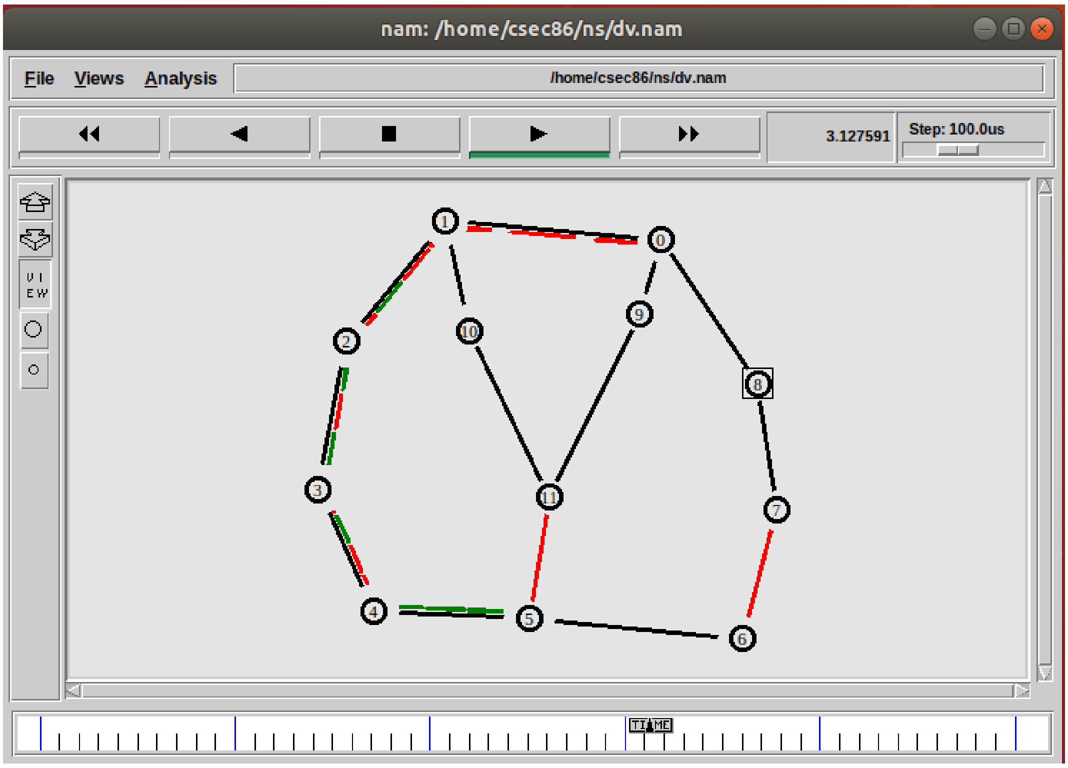
OUTPUT:

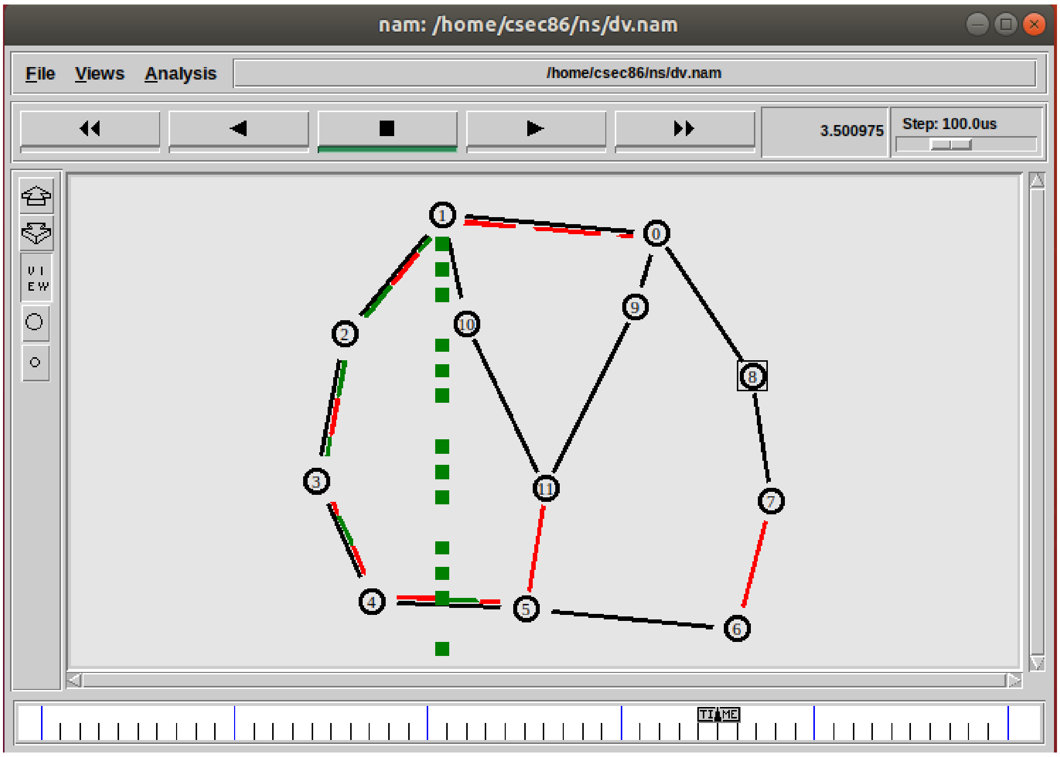


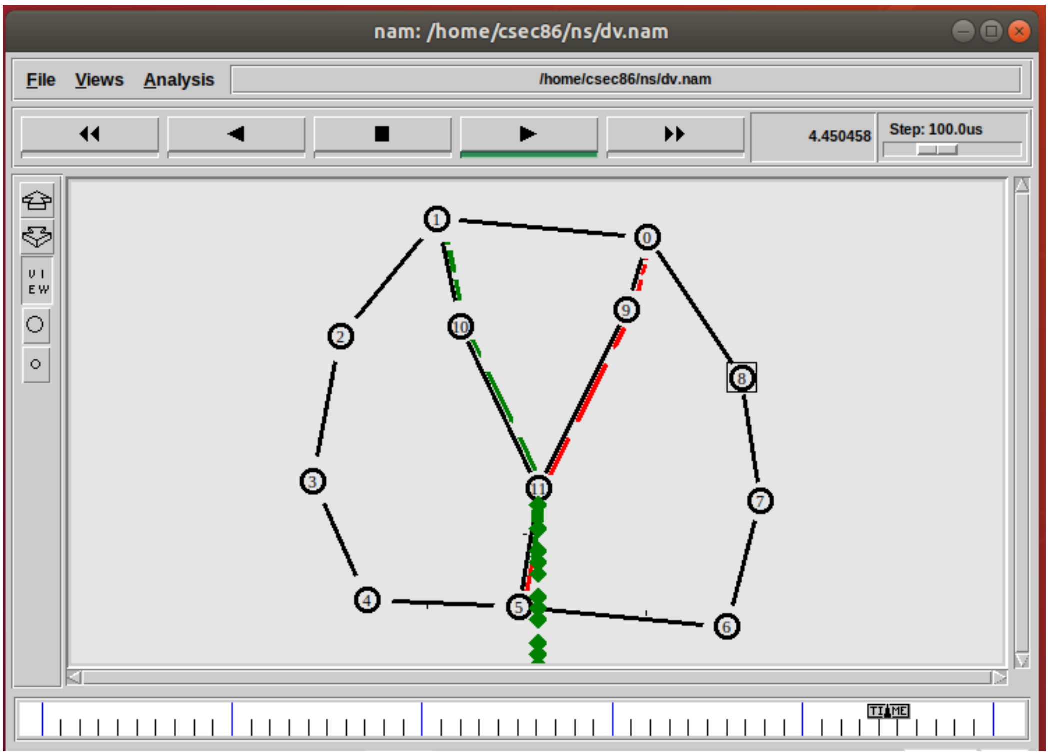

nam: /home/csec86/ns/dv.nam
File
Views
Analysis
/home/csec86/ns/dv.nam
1.471439
Step: 100.0us
TIME
0
1
2
3
4
5
6
7
8
9
10
11


nam: /home/csec86/ns/dv.nam
File
Views
Analysis
/home/csec86/ns/dv.nam
2.238938
Step: 100.0us
TIME
0
1
2
3
4
5
6
7
8
9
10
11


nam: /home/csec86/ns/dv.nam
File
Views
Analysis
/home/csec86/ns/dv.nam
2.454775
Step: 100.0us
VIEW
TIME
1
0
2
10
9
8
3
11
7
4
5
6


nam: /home/csec86/ns/dv.nam
File
Views
Analysis
/home/csec86/ns/dv.nam
3.127591
Step: 100.0us
VIEW
TIME
1
0
2
10
9
8
3
11
7
4
5
6


nam: /home/csec86/ns/dv.nam
File
Views
Analysis
/home/csec86/ns/dv.nam
3.500975
Step: 100.0us
VIEW
1
0
2
10
9
8
3
11
7
4
5
6
TIME

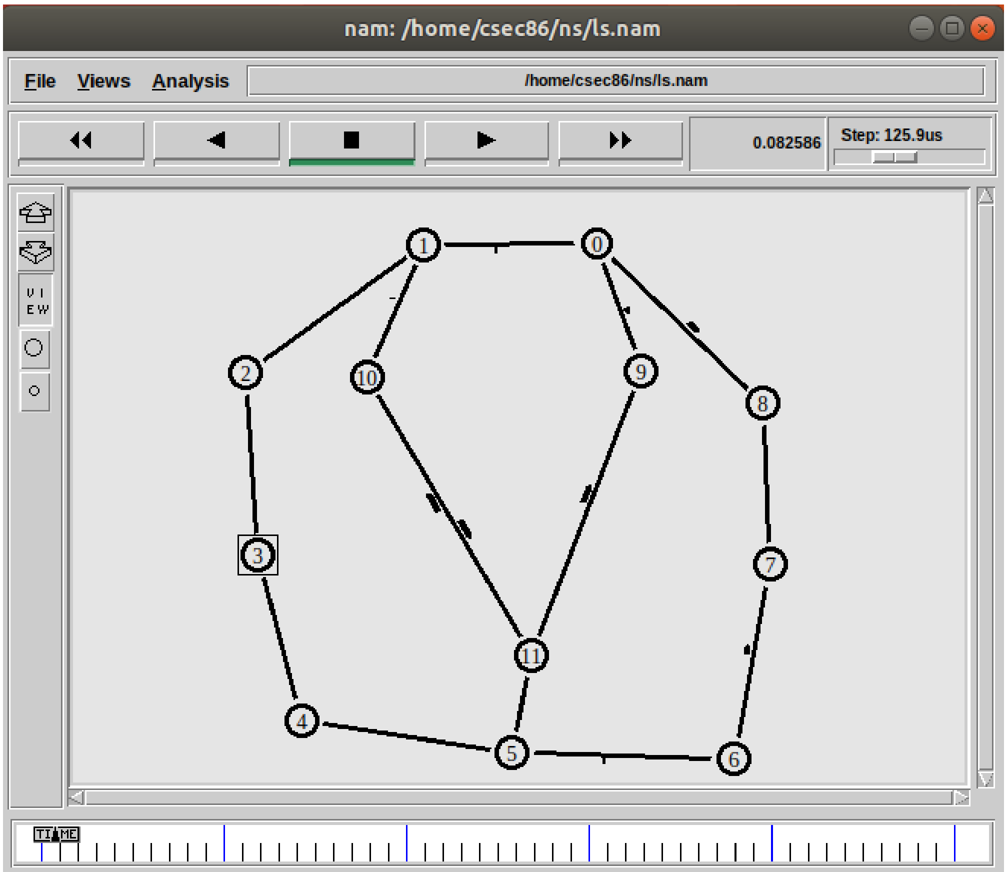

nam: /home/csec86/ns/dv.nam
File
Views
Analysis
/home/csec86/ns/dv.nam
4.450458
Step: 100.0us
VIEW
1
0
2
10
9
8
3
11
7
4
5
6
TIME

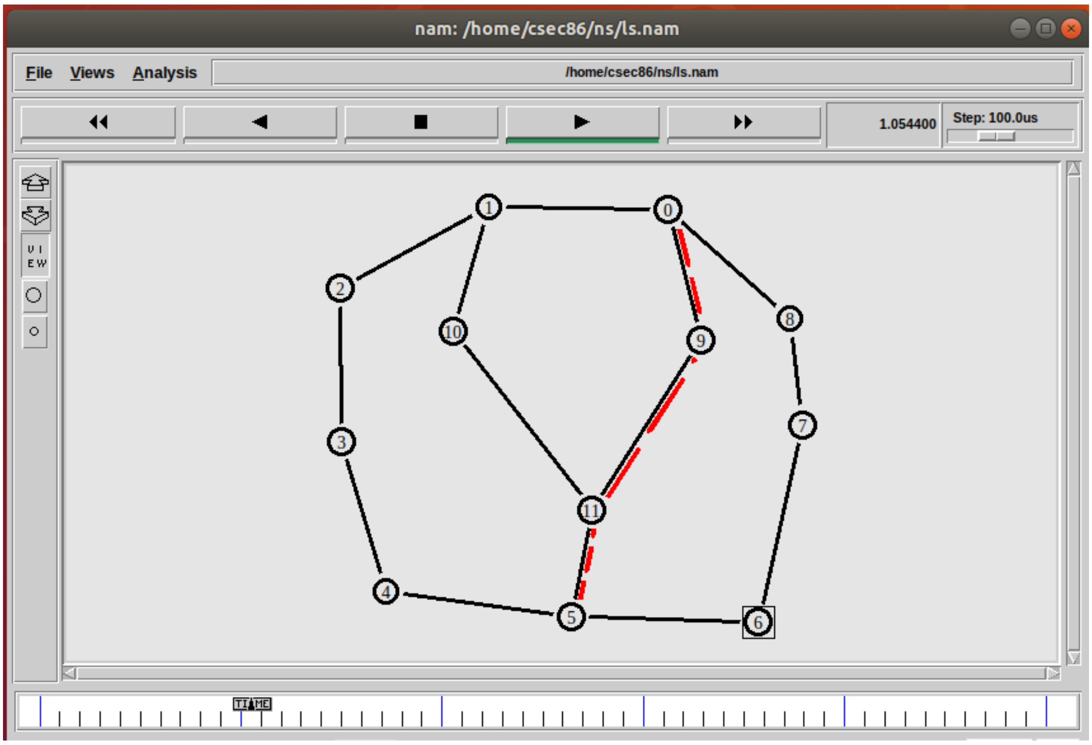
PROGRAM CODE:

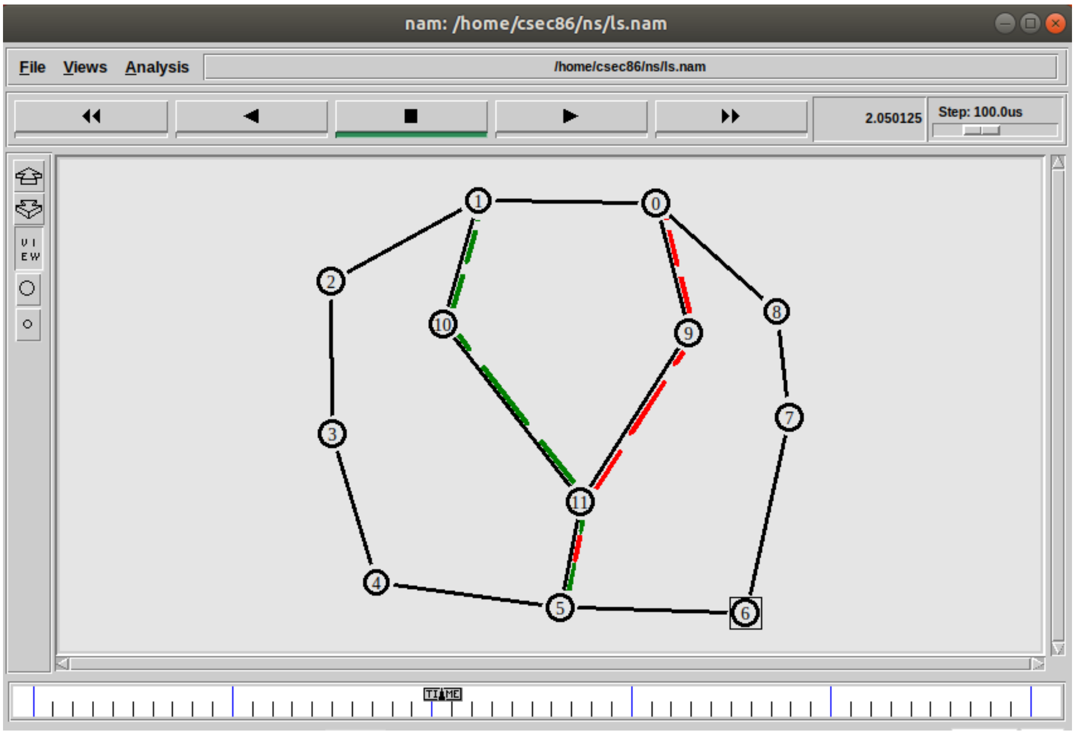
```
set ns [new Simulator]
set nr [open ls.tr w]
$ns trace-all $nr
set nf [open ls.nam w]
$ns namtrace-all $nf
proc finish { } {
global ns nr nf
$ns flush-trace
close $nf
close $nr
exec nam ls.nam &
exit 0
}

for { set i 0 } { $i < 12} { incr i 1 } {
set n($i) [$ns node]}

for {set i 0} {$i < 8} {incr i} {
$ns duplex-link $n($i) $n([expr $i+1]) 1Mb 10ms DropTail }
$ns duplex-link $n(0) $n(8) 1Mb 10ms DropTail
$ns duplex-link $n(1) $n(10) 1Mb 10ms DropTail
$ns duplex-link $n(0) $n(9) 1Mb 10ms DropTail
$ns duplex-link $n(9) $n(11) 1Mb 10ms DropTail
$ns duplex-link $n(10) $n(11) 1Mb 10ms DropTail
$ns duplex-link $n(11) $n(5) 1Mb 10ms DropTail

set udp0 [new Agent/UDP]
$ns attach-agent $n(0) $udp0
set cbr0 [new Application/Traffic/CBR]
$cbr0 set packetSize_ 500
$cbr0 set interval_ 0.005
$cbr0 attach-agent $udp0
set null0 [new Agent/Null]
$ns attach-agent $n(5) $null0
$ns connect $udp0 $null0

set udp1 [new Agent/UDP]
$ns attach-agent $n(1) $udp1
set cbr1 [new Application/Traffic/CBR]
$cbr1 set packetSize_ 500
$cbr1 set interval_ 0.005
$cbr1 attach-agent $udp1
set null0 [new Agent/Null]
$ns attach-agent $n(5) $null0
```

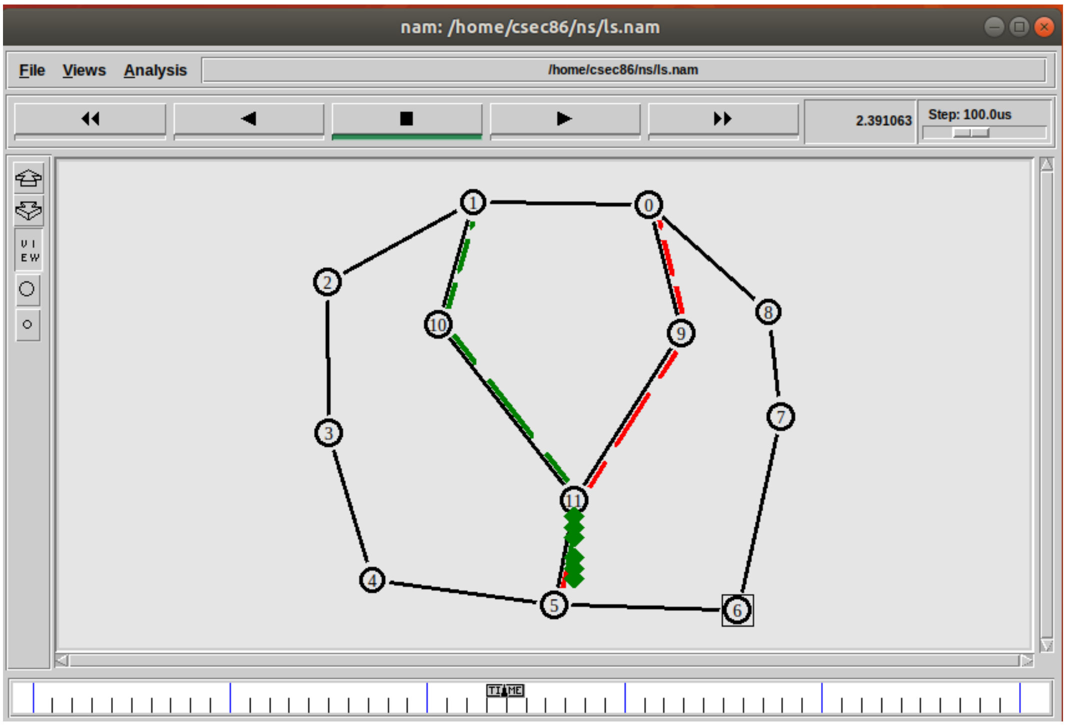
```
$ns connect $udp1 $null0

$ns rtproto LS

$ns rtmodel-at 3.0 down $n(11) $n(5)
$ns rtmodel-at 3.0 down $n(7) $n(6)
$ns rtmodel-at 4.0 up $n(11) $n(5)
$ns rtmodel-at 4.0 up $n(7) $n(6)

$udp0 set fid_ 1
$udp1 set fid_ 2

$ns color 1 Red
$ns color 2 Green

$ns at 1.0 "$cbr0 start"
$ns at 2.0 "$cbr1 start"

$ns at 5 "finish"
$ns run
```

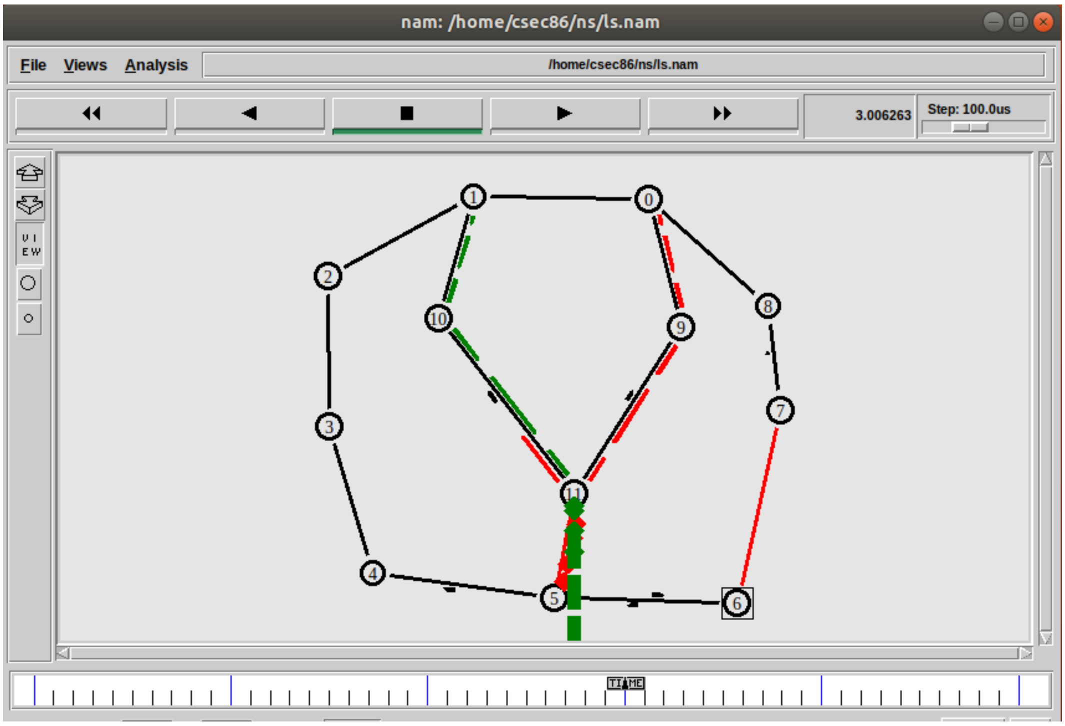
OUTPUT:

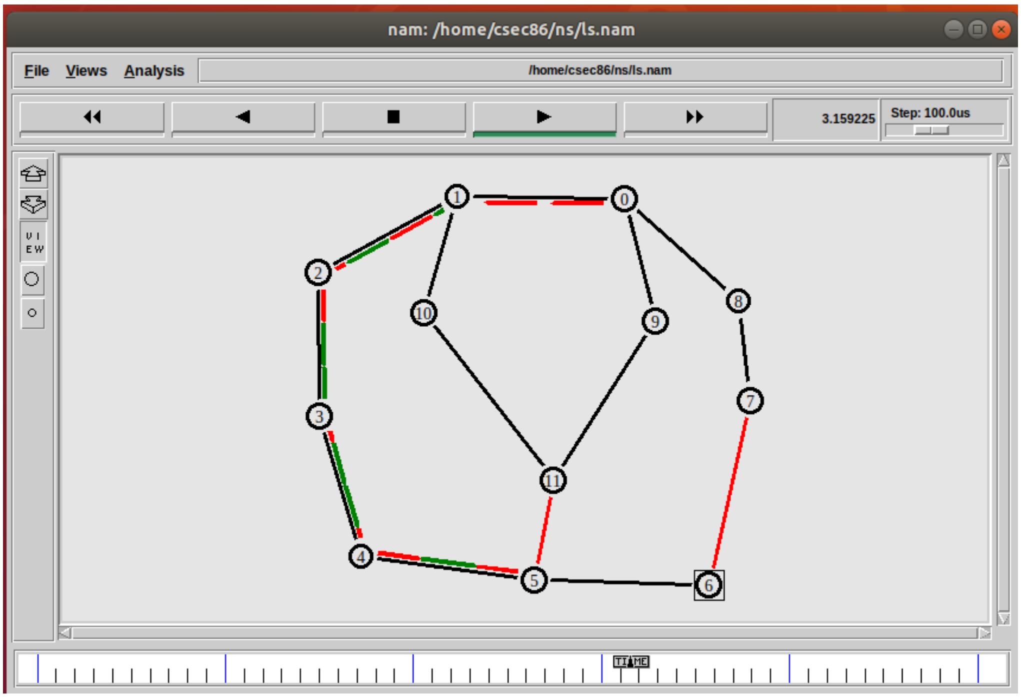



nam: /home/csec86/ns/ls.nam
File Views Analysis
/home/csec86/ns/ls.nam
1.054400
Step: 100.0us
0
1
2
3
4
5
6
7
8
9
10
11
TIME

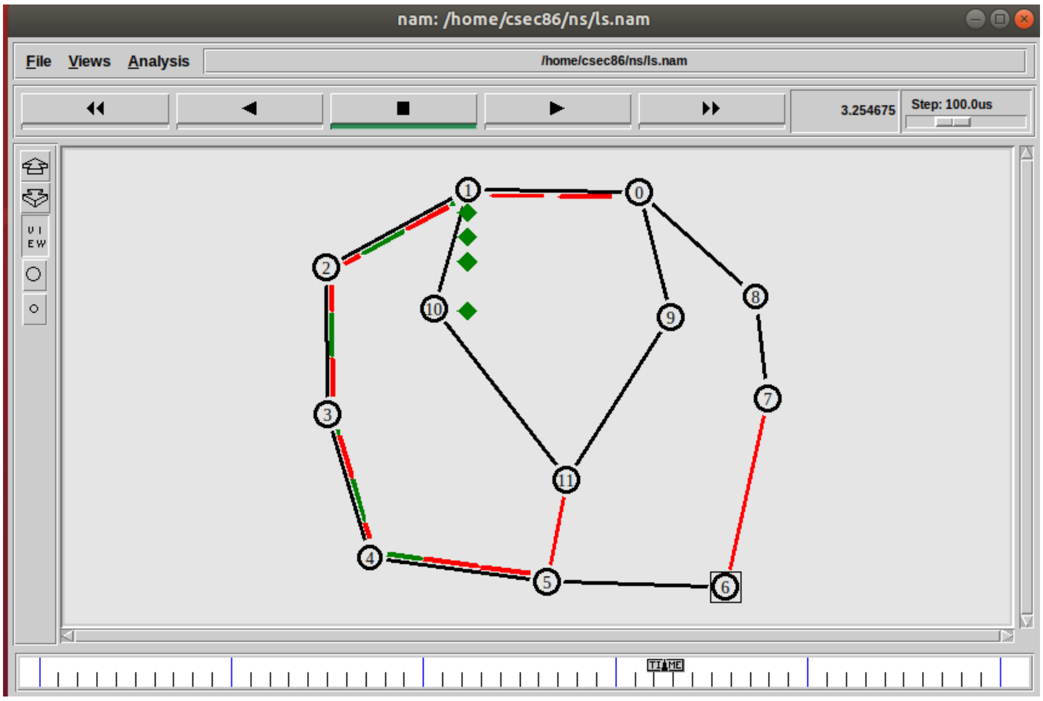

nam: /home/csec86/ns/ls.nam
File Views Analysis
/home/csec86/ns/ls.nam
2.050125
Step: 100.0us
0
1
2
3
4
5
6
7
8
9
10
11
TIME


nam: /home/csec86/ns/ls.nam
File
Views
Analysis
/home/csec86/ns/ls.nam
2.391063
Step: 100.0us
VIEW
0
1
2
3
4
5
6
7
8
9
10
11
TIME

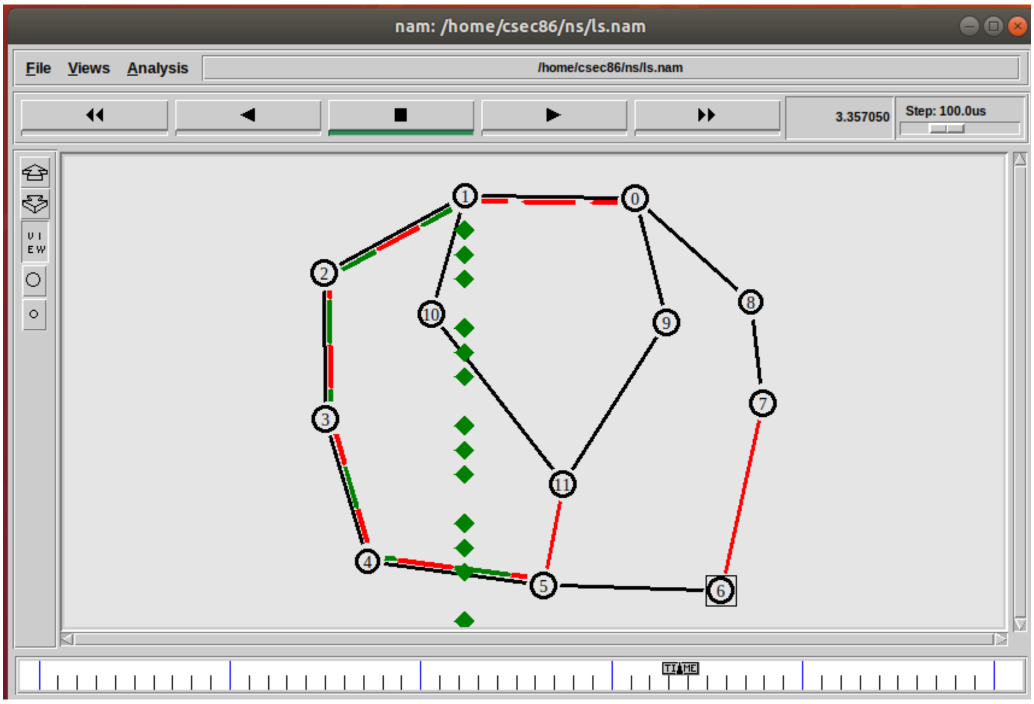

nam: /home/csec86/ns/ls.nam
File
Views
Analysis
/home/csec86/ns/ls.nam
3.006263
Step: 100.0us
VIEW
0
1
2
3
4
5
6
7
8
9
10
11
TIME



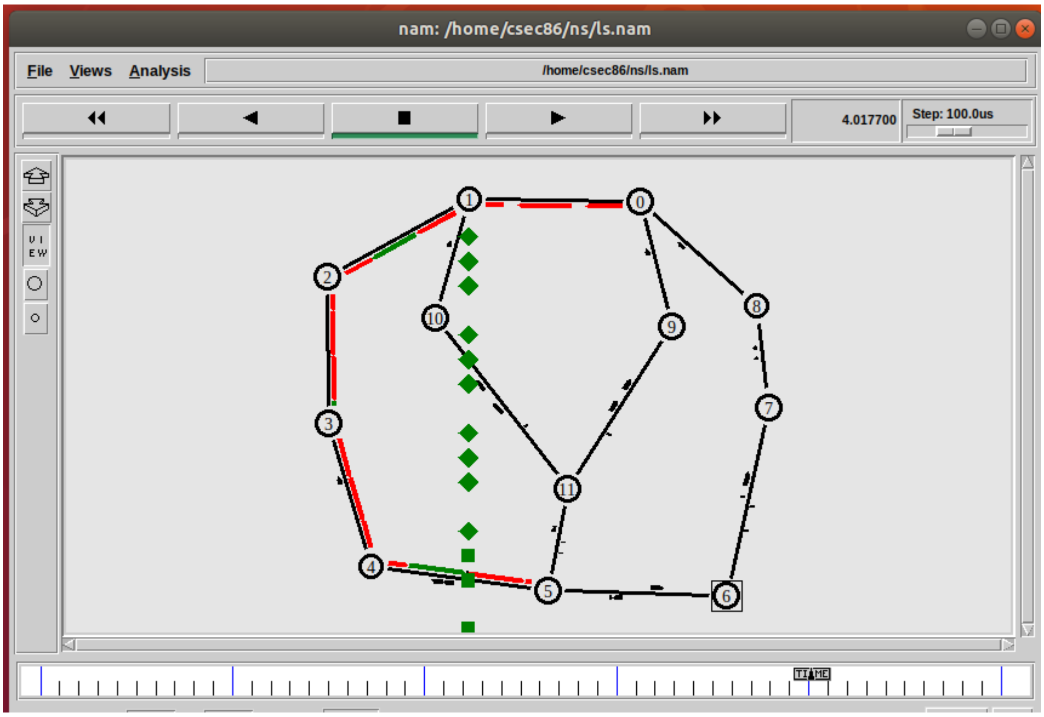
nam: /home/csec86/ns/ls.nam
File
Views
Analysis
/home/csec86/ns/ls.nam
3.159225
Step: 100.0us
0
1
2
3
4
5
6
7
8
9
10
11
TIME



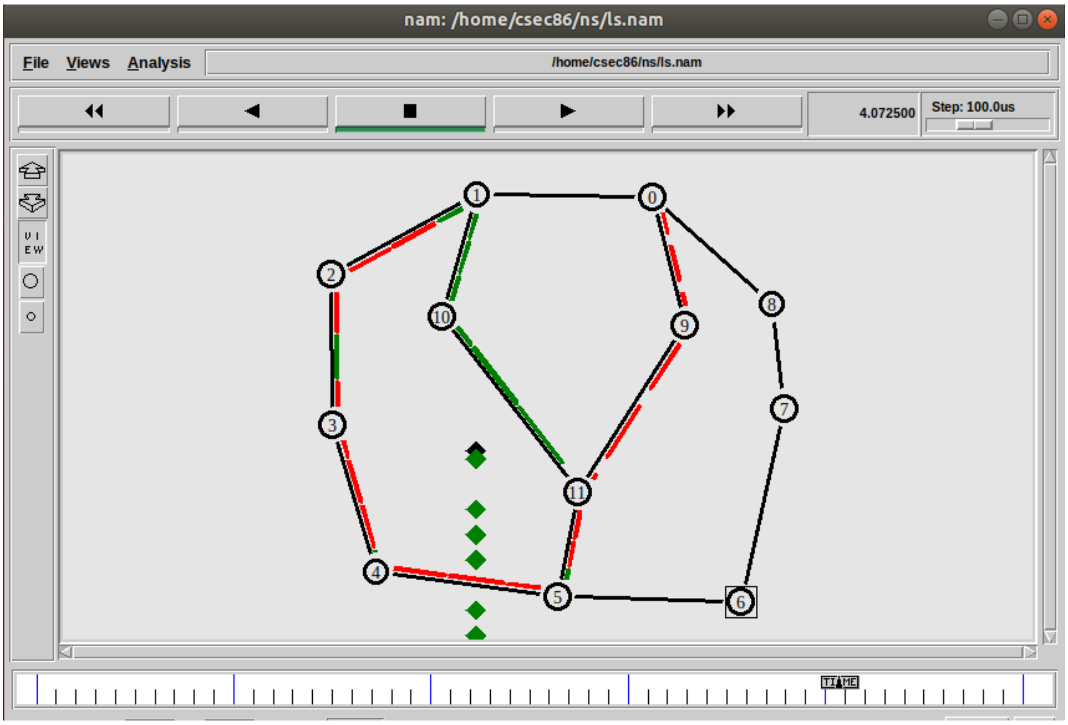
nam: /home/csec86/ns/ls.nam
File
Views
Analysis
/home/csec86/ns/ls.nam
3.254675
Step: 100.0us
0
1
2
3
4
5
6
7
8
9
10
11
TIME

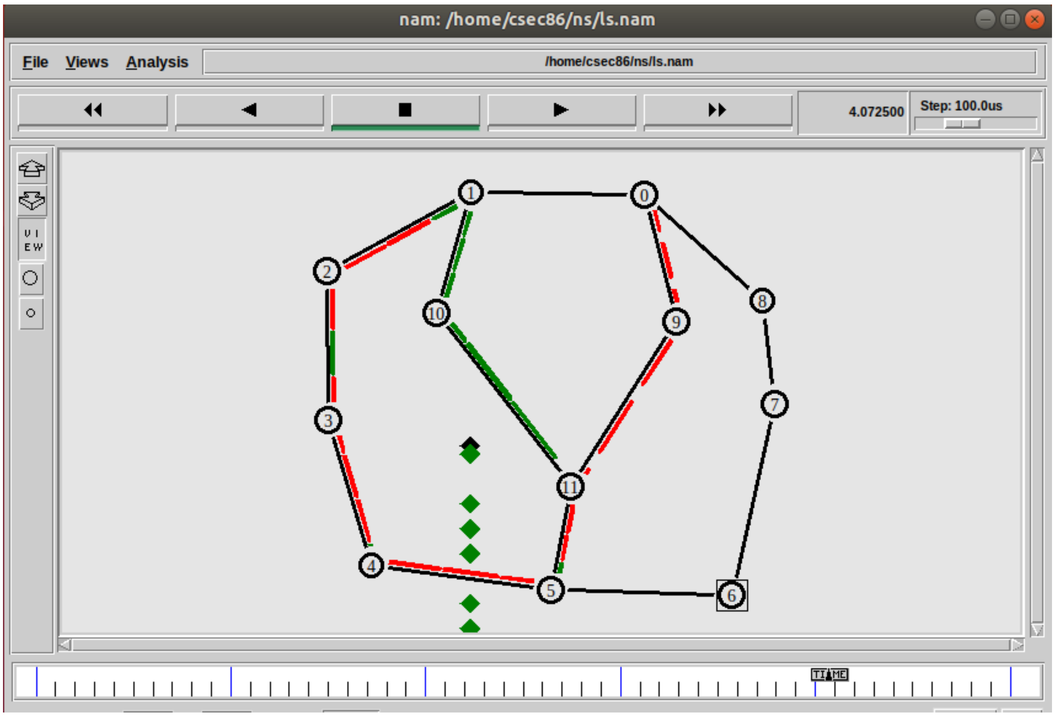


nam: /home/csec86/ns/ls.nam
File
Views
Analysis
/home/csec86/ns/ls.nam
3.357050
Step: 100.0us
0
1
2
3
4
5
6
7
8
9
10
11
TIME



nam: /home/csec86/ns/ls.nam
File
Views
Analysis
/home/csec86/ns/ls.nam
4.017700
Step: 100.0us
0
1
2
3
4
5
6
7
8
9
10
11
TIME

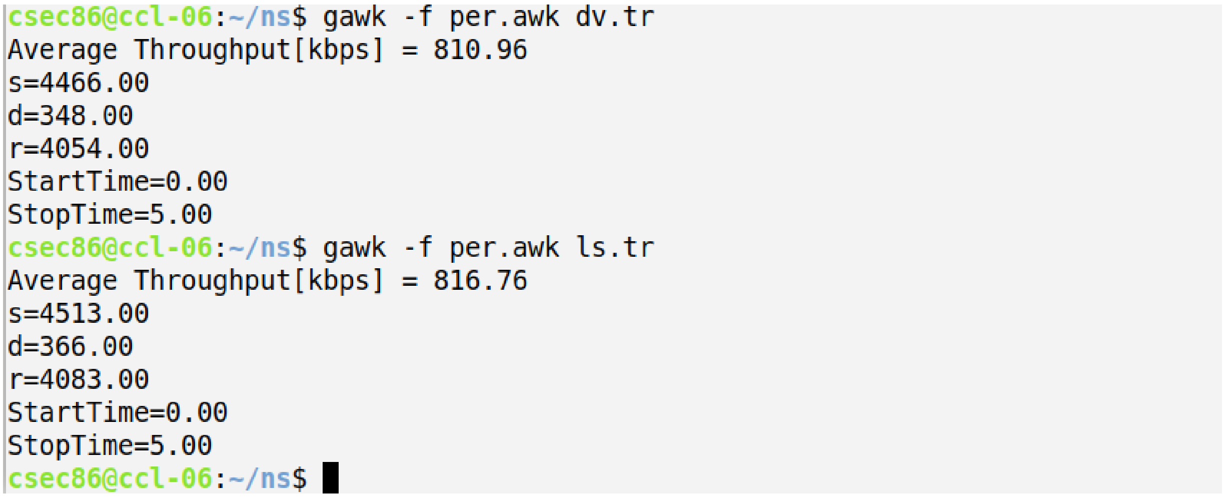

nam: /home/csec86/ns/ls.nam
File
Views
Analysis
/home/csec86/ns/ls.nam
4.072500
Step: 100.0us
V I E W
0
1
2
3
4
5
6
7
8
9
10
11
TIME


nam: /home/csec86/ns/ls.nam
File
Views
Analysis
/home/csec86/ns/ls.nam
4.072500
Step: 100.0us
V I E W
0
1
2
3
4
5
6
7
8
9
10
11
TIME

PROGRAM CODE:

```
BEGIN {
recvdSize = 0
txsize=0
drpSize=0
startTime = 0
stopTime = 0
thru=0
}
{
event = $1
time = $2
node_id = $3
pkt_size = $6
level = $5
# Store start time
if (level == "cbr" && (event == "+" || event == "s") )
{
if (time < startTime)
{
startTime = time
}
txsize++;}
# Update total received packetsâ€TM size and store packets arrival time
if (level == "cbr" && event == "r" )
{
if (time > stopTime)
{
stopTime = time
}
recvdSize++
}
if (level == "cbr" && event == "d" )
{
drpSize++
}
}
END {
printf("Average Throughput[kbps] =
%.2f\ns=%.2f\nd=%.2f\nr=%.2f\nStartTime=%.2f\nStopTime=%.2f\n",(recvdSize/(stopTi
me-startTime)),txsize,drpSize,recvdSize,startTime,stopTime)
}
```

OUTPUT:



```
csec86@ccl-06:~/ns$ gawk -f per.awk dv.tr
Average Throughput[kbps] = 810.96
s=4466.00
d=348.00
r=4054.00
StartTime=0.00
StopTime=5.00
csec86@ccl-06:~/ns$ gawk -f per.awk ls.tr
Average Throughput[kbps] = 816.76
s=4513.00
d=366.00
r=4083.00
StartTime=0.00
StopTime=5.00
csec86@ccl-06:~/ns$ 
```