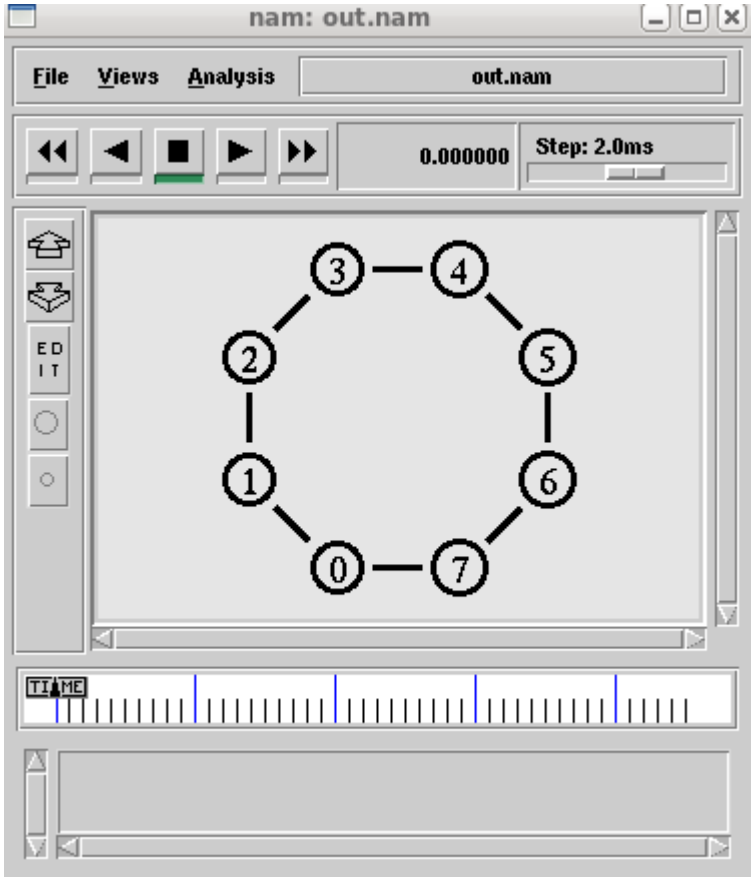
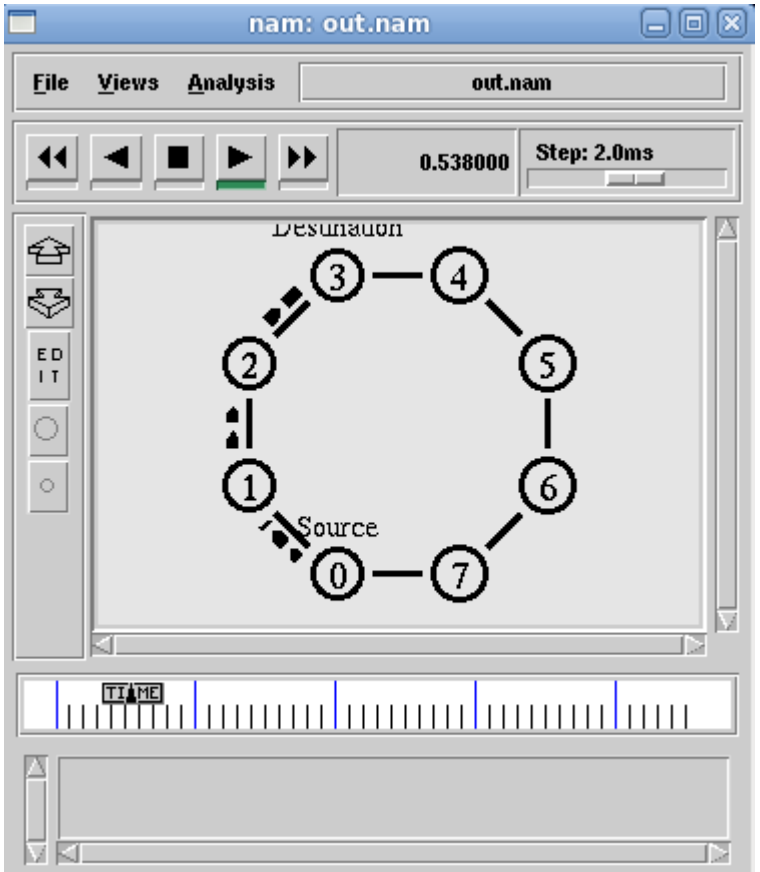
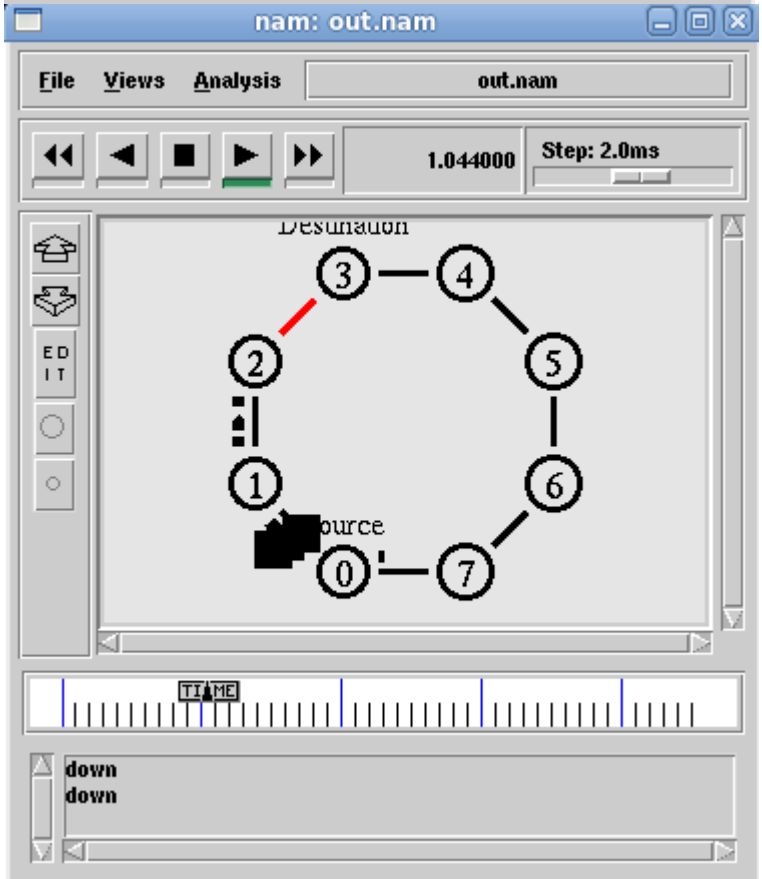
```
dvr 1
set ns [new Simulator]
#Use distance vector routing
$ns rtproto DV
set nf [open out.nam w]
$ns namtrace-all $nf
# Open tracefile
set nt [open trace.tr w]
$ns trace-all $nt
#Define 'finish' procedure
proc finish {} {
global ns nf
$ns flush-trace
#Close the trace file
close $nf
#Execute nam on the trace file
exec nam -a out.nam &
exit 0
}
# Create 8 nodes
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
set n6 [$ns node]
set n7 [$ns node]
set n8 [$ns node]
# Specify link characterestics
$ns duplex-link $n1 $n2 1Mb 10ms DropTail
$ns duplex-link $n2 $n3 1Mb 10ms DropTail
$ns duplex-link $n3 $n4 1Mb 10ms DropTail
$ns duplex-link $n4 $n5 1Mb 10ms DropTail
$ns duplex-link $n5 $n6 1Mb 10ms DropTail
$ns duplex-link $n6 $n7 1Mb 10ms DropTail
$ns duplex-link $n7 $n8 1Mb 10ms DropTail
$ns duplex-link $n8 $n1 1Mb 10ms DropTail
# specify layout as a octagon
$ns duplex-link-op $n1 $n2 orient left-up
$ns duplex-link-op $n2 $n3 orient up
$ns duplex-link-op $n3 $n4 orient right-up
$ns duplex-link-op $n4 $n5 orient right
$ns duplex-link-op $n5 $n6 orient right-down
$ns duplex-link-op $n6 $n7 orient down
$ns duplex-link-op $n7 $n8 orient left-down
$ns duplex-link-op $n8 $n1 orient left
#Create a UDP agent and attach it to node n1
set udp0 [new Agent/UDP]
$ns attach-agent $n1 $udp0
#Create a CBR traffic source and attach it to udp0
set cbr0 [new Application/Traffic/CBR]
```

```
$cbr0 set packetSize_ 500
$cbr0 set interval_ 0.005
$cbr0 attach-agent $udp0
#Create a Null agent (a traffic sink) and attach it to node n

set null0 [new Agent/Null]
$ns attach-agent $n4 $null0
#Connect the traffic source with the traffic sink
$ns connect $udp0 $null0
#Schedule events for the CBR agent and the network dynamics
$ns at 0.0 "$n1 label Source"
$ns at 0.0 "$n4 label Destination"
$ns at 0.5 "$cbr0 start"
$ns rtmodel-at 1.0 down $n3 $n4
$ns rtmodel-at 2.0 up $n3 $n4
$ns at 4.5 "$cbr0 stop"
#Call the finish procedure after 5 seconds of simulation tim

$ns at 5.0 "finish"
#Run the simulation
$ns run
```



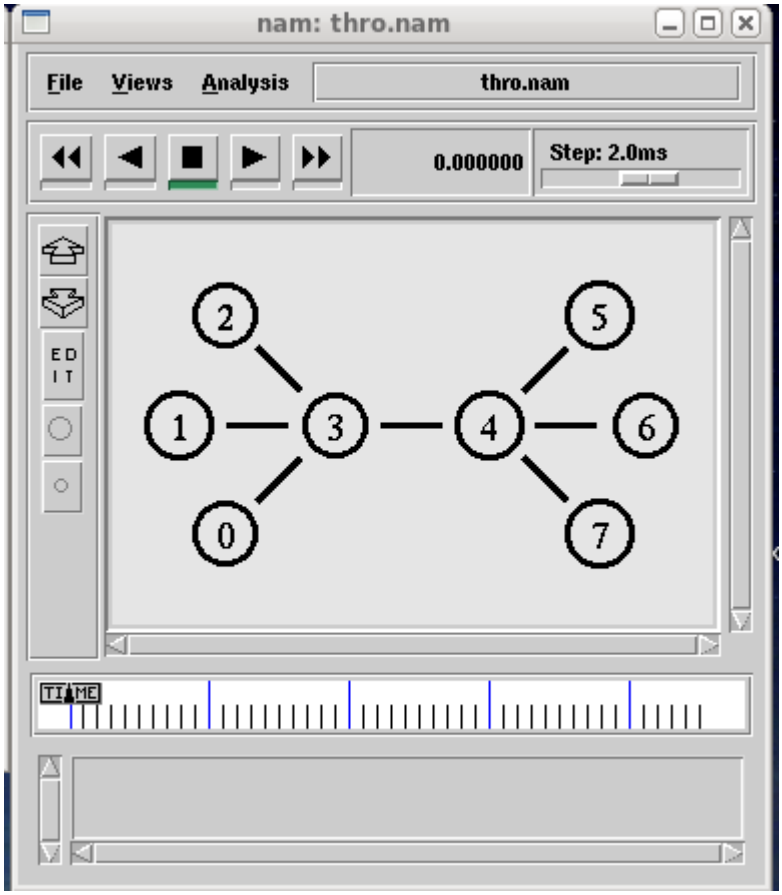

nam: out.nam
File
Views
Analysis
out.nam
0.538000
Step: 2.0ms
Destination
3
4
2
5
1
6
Source
0
7
TIME

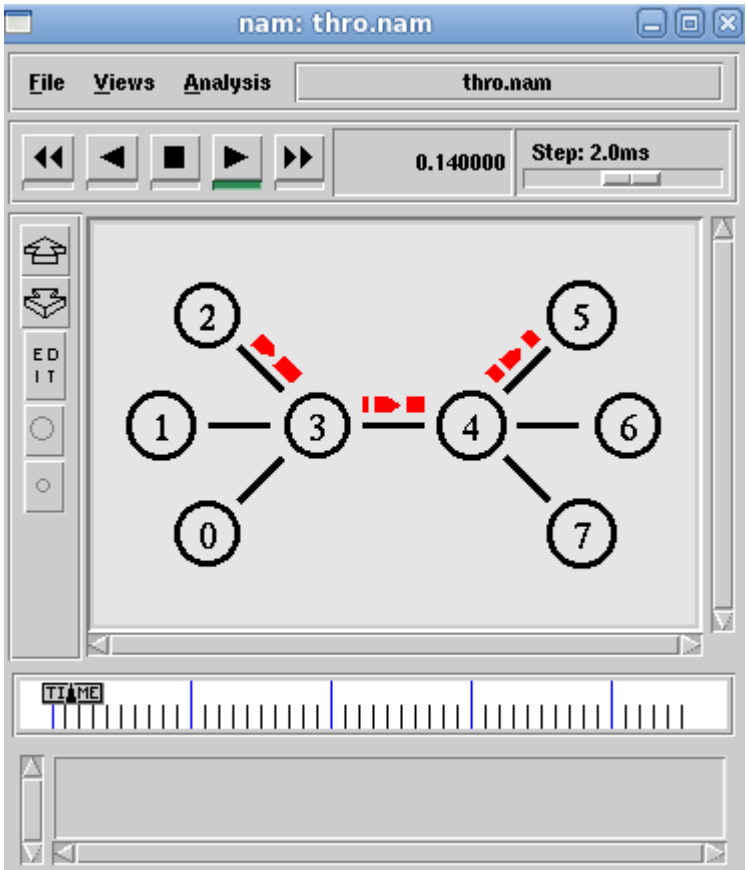

nam: out.nam
File
Views
Analysis
out.nam
1.044000
Step: 2.0ms
Destination
3
4
2
5
1
6
0
7
TIME
down
down

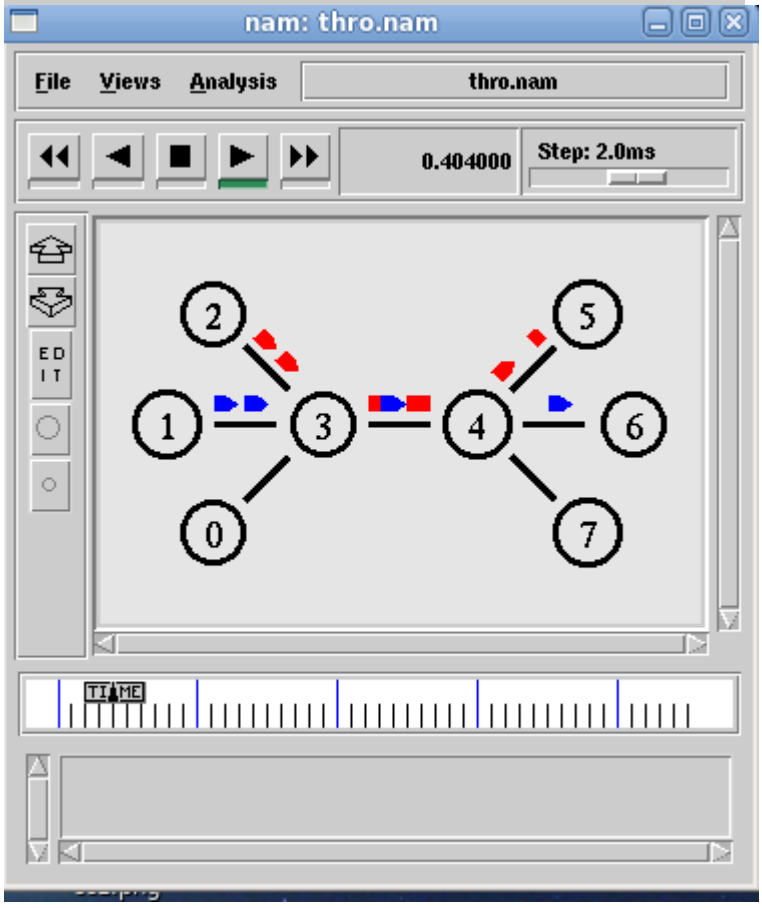
```
first.tcl
set ns [new Simulator]
set nr [open thro red.tr w]
$ns trace-all $nr
set nf [open thro.nam w]
$ns namtrace-all $nf
proc finish { } {
global ns nr nf
$ns flush-trace
close $nf
close $nr
exec nam thro.nam &
exit 0 }

set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
set n6 [$ns node]
set n7 [$ns node]
$ns duplex-link $n0 $n3 1Mb 10ms RED
$ns duplex-link $n1 $n3 1Mb 10ms RED
$ns duplex-link $n2 $n3 1Mb 10ms RED
$ns duplex-link $n3 $n4 1Mb 10ms RED
$ns duplex-link $n4 $n5 1Mb 10ms RED
$ns duplex-link $n4 $n6 1Mb 10ms RED
$ns duplex-link $n4 $n7 1Mb 10ms RED
$ns duplex-link-op $n0 $n3 orient right-up
$ns duplex-link-op $n3 $n4 orient middle
$ns duplex-link-op $n2 $n3 orient right-down
$ns duplex-link-op $n4 $n5 orient right-up
$ns duplex-link-op $n4 $n7 orient right-down
$ns duplex-link-op $n1 $n3 orient right
$ns duplex-link-op $n6 $n4 orient left
set udp0 [new Agent/UDP]
$ns attach-agent $n2 $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 $n5 $null0
$ns connect $udp0 $null0
set udp1 [new Agent/UDP]
$ns attach-agent $n1 $udp1
set cbr1 [new Application/Traffic/CBR]
$cbr1 set packetSize_ 500
$cbr1 set interval_ 0.005
$cbr1 attach-agent $udp1
```

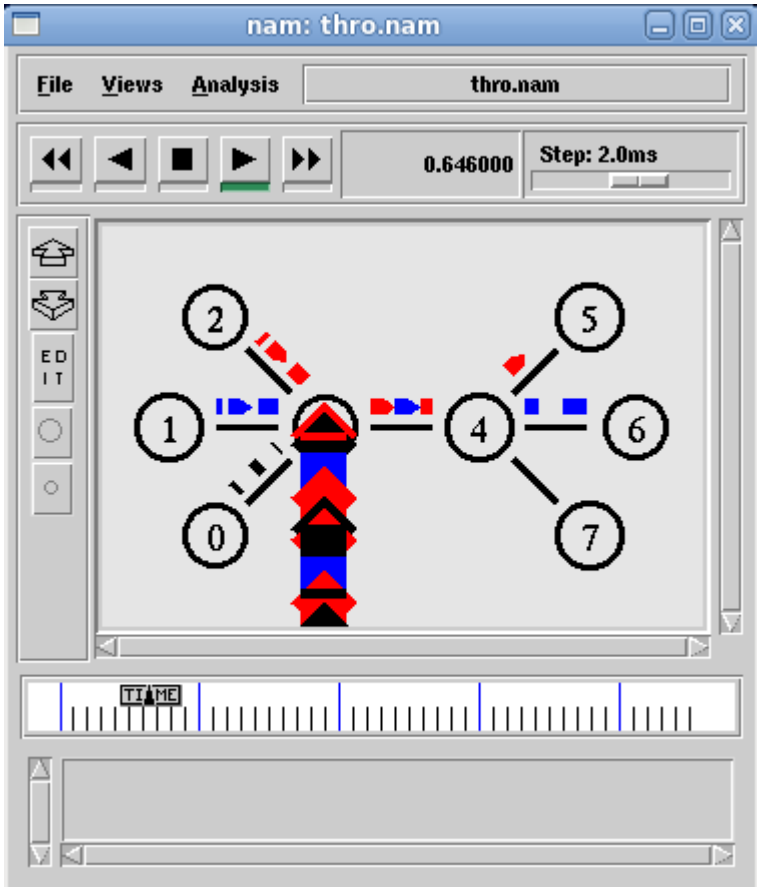
```
set null0 [new Agent/Null]
$ns attach-agent $n6 $null0
$ns connect $udp1 $null0
set udp2 [new Agent/UDP]
$ns attach-agent $n0 $udp2
set cbr2 [new Application/Traffic/CBR]
$cbr2 set packet size_ 500
$cbr2 set interval_ 0.005
$cbr2 attach-agent $udp2
set null0 [new Agent/Null]
$ns attach-agent $n7 $null0
$ns connect $udp2 $null0
$udp0 set fid_ 1

$udp1 set fid_ 2
$udp2 set fid_ 3
$ns color 1 Red
$ns color 2 Green
$ns color 2 Blue
$ns at 0.1 "$cbr0 start"
$ns at 0.2 "$cbr1 start"
$ns at 0.5 "$cbr2 start"
$ns at 4.0 "$cbr2 stop"
$ns at 4.2 "$cbr1 stop"
$ns at 4.5 "$cbr0 stop"
$ns at 5.0 "finish"
```

OUTPUT


nam: thro.nam
File
Views
Analysis
thro.nam
0.000000
Step: 2.0ms
ED
IT
2
5
1
3
4
6
0
7
TIME


nam: thro.nam
File
Views
Analysis
thro.nam
0.140000
Step: 2.0ms
ED
IT
2
5
1
3
4
6
0
7
TIME

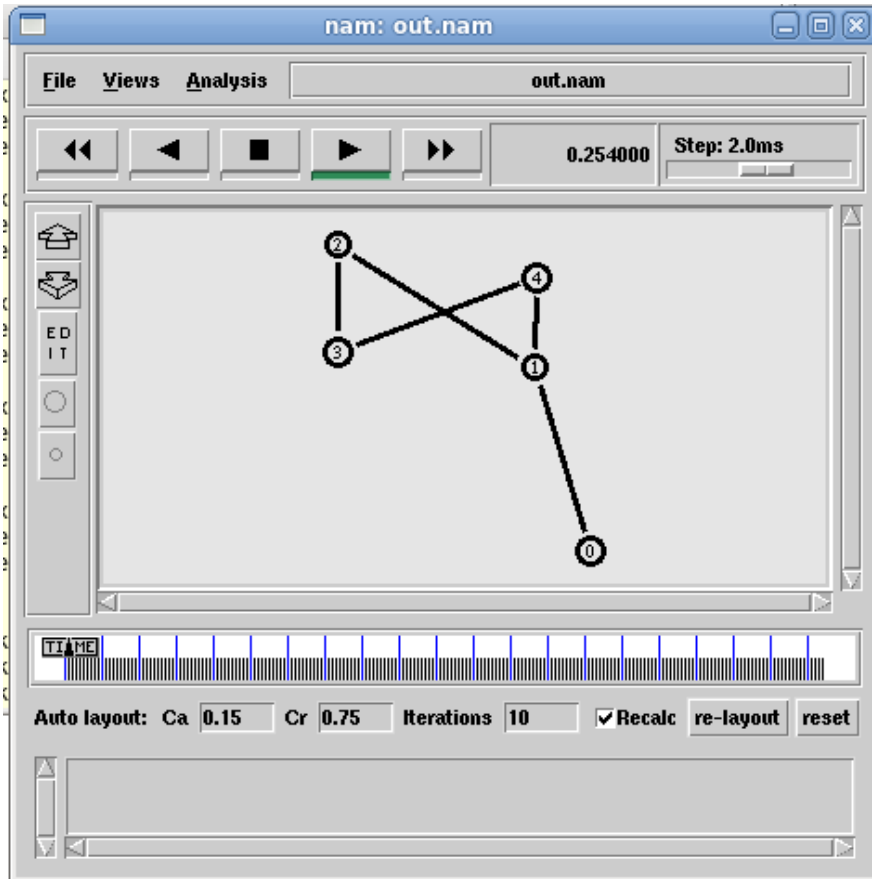

nam: thro.nam
File
Views
Analysis
thro.nam
0.404000
Step: 2.0ms
ED
IT
2
5
1
3
4
6
0
7
TIME

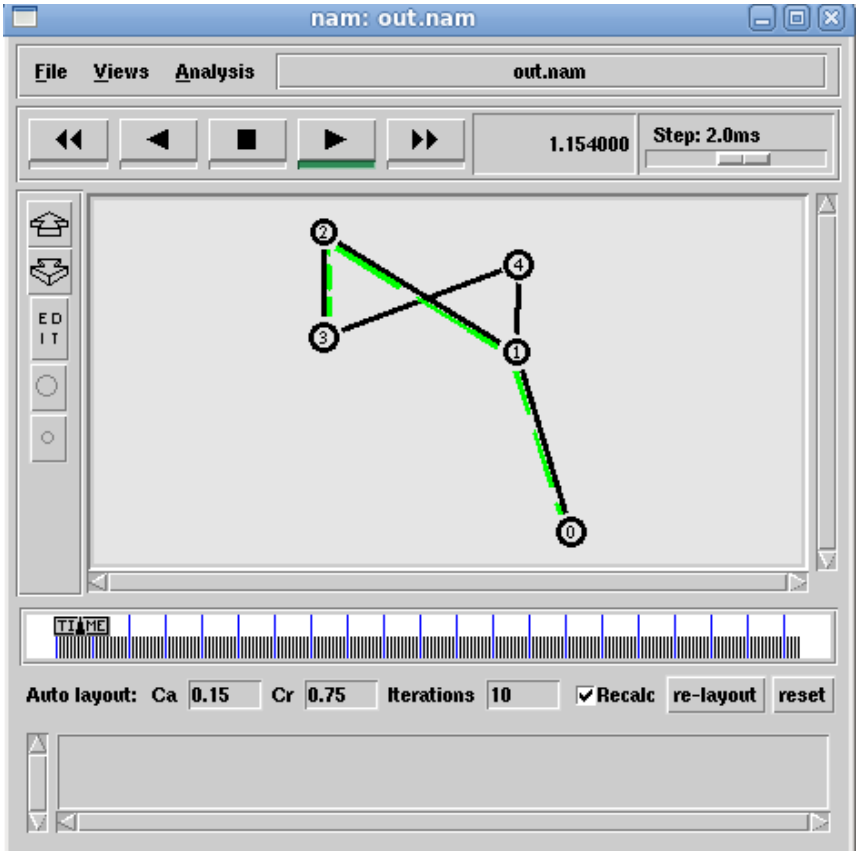

nam: thro.nam
File
Views
Analysis
thro.nam
0.646000
Step: 2.0ms
EDIT
TIME
0
1
2
4
5
6
7

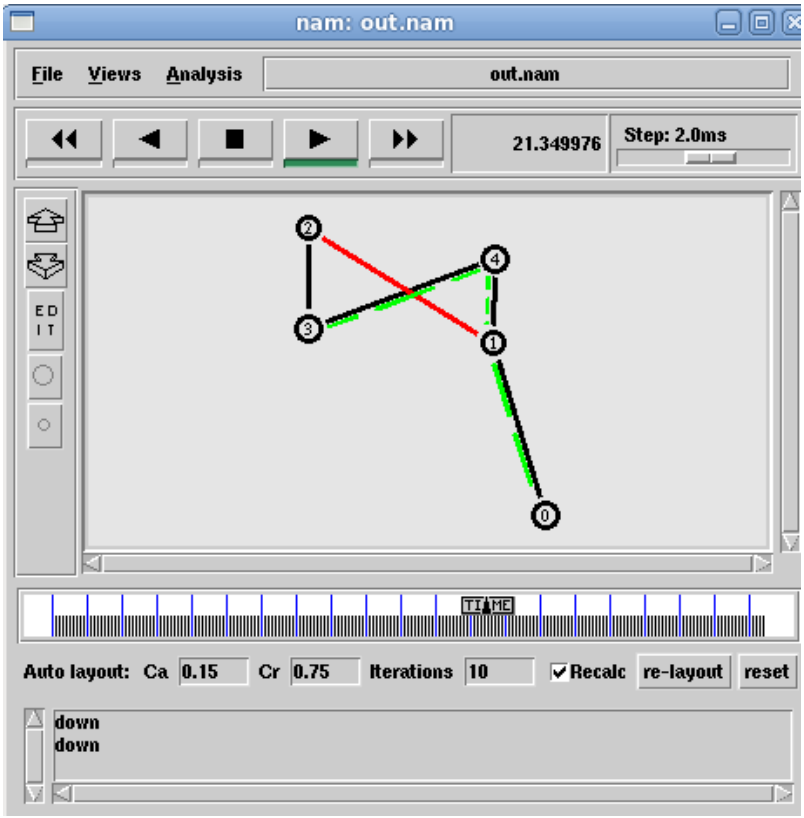
lsr.tcl

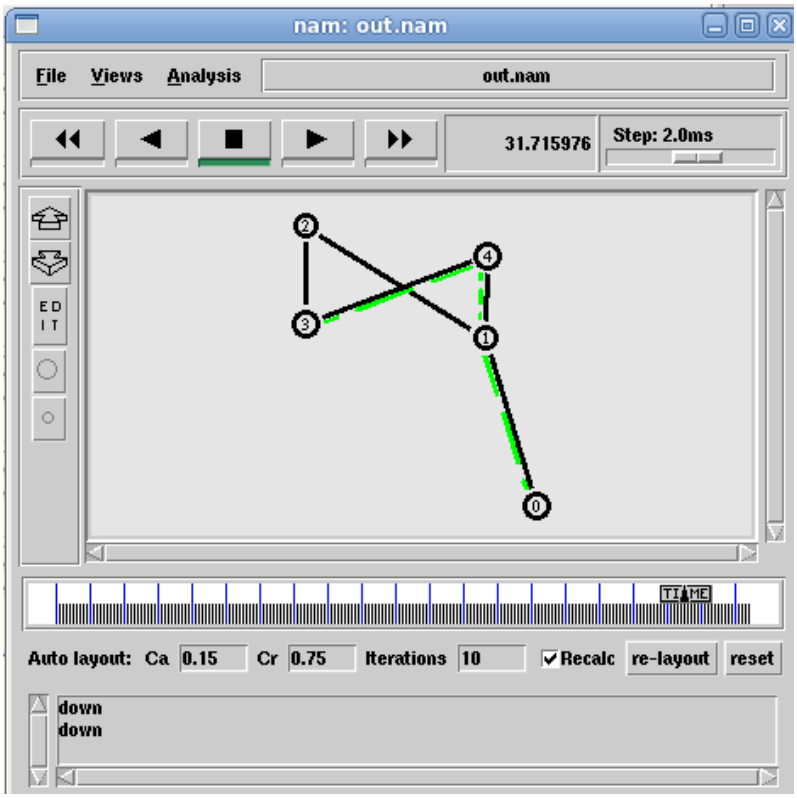
```
#Create a simulator object
set ns [new Simulator]
#Define different colors for data flows(for NAM)
$ns color 1 Blue
$ns color 2 Red
#Open the NAM trace file
set nf [open out.nam w]
$ns namtrace-all $nf
#Define a 'finish' procedure
proc finish { } {
    global ns nf
    $ns flush-trace
    #close the NAM trace file
    close $nf
    #Execute NAM on the trace file
     exec nam out.nam &
      exit 0
}
for {set i 0} {$i < 5} {incr i 1} {
    set n($i) [$ns node]}
for {set i 0} {$i < 4} {incr i} {
$ns duplex-link $n($i) $n([expr $i+1]) 1Mb 10ms DropTail}
$ns duplex-link $n(0) $n(1) 1Mb 10ms DropTail
$ns duplex-link $n(1) $n(2) 1Mb 10ms DropTail
$ns duplex-link $n(2) $n(3) 1Mb 10ms DropTail
$ns duplex-link $n(3) $n(4) 1Mb 10ms DropTail
$ns duplex-link $n(4) $n(1) 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(3) $null0
$ns connect $udp0 $null0
$ns rtproto DV
$ns rtmodel-at 15.0 down $n(1) $n(2)
$ns rtmodel-at 25.0 up $n(1) $n(2)
$udp0 set fid_ 1
$ns color 1 Green
$ns at 1.0 "$cbr0 start"
$ns at 35 "finish"
#Run the simulation
$ns run
```

OUTPUT


nam: out.nam
File
Views
Analysis
out.nam
0.254000
Step: 2.0ms
ED
IT
TIME
Auto layout:
Ca
0.15
Cr
0.75
Iterations
10
Recalc
re-layout
reset

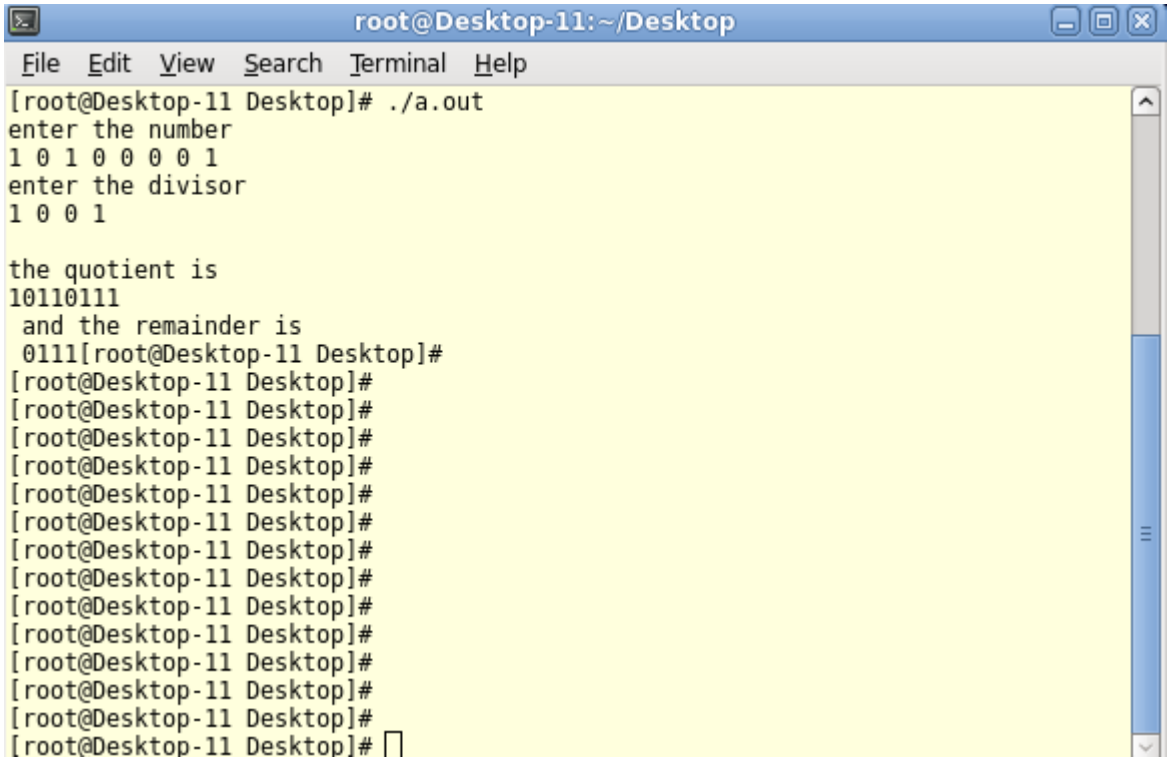

nam: out.nam
File
Views
Analysis
out.nam
1.154000
Step: 2.0ms
ED
IT
TIME
Auto layout:
Ca
0.15
Cr
0.75
Iterations
10
Recalc
re-layout
reset


nam: out.nam
File
Views
Analysis
out.nam
21.349976
Step: 2.0ms
ED
IT
TIME
Auto layout:
Ca
0.15
Cr
0.75
Iterations
10
Recalc
re-layout
reset
down
down


nam: out.nam
File
Views
Analysis
out.nam
31.715976
Step: 2.0ms
ED
IT
TIME
Auto layout:
Ca
0.15
Cr
0.75
Iterations
10
Recalc
re-layout
reset
down
down

```
#include<stdio.h>
void main()
{
int i,f[20],n[50],div[50],j,temp,quotient[20],z[10];
printf("enter the number\n");
for(i=0;i<8;i++)
{
scanf("%d",&n[i]);
}
printf("enter the divisor\n");
for(i=0;i<4;i++)
{
scanf("%d",&div[i]);
}

for(i=8;i<12;i++)
{
n[i]=0;
}

for(i=0;i<8;i++)
{
temp=i;
if(n[i]==1)
{
for (j=0;j<4;j++)
{
if (n[temp]==div[j])
{n[temp]=0;
f[j]=0;}
else
{n[temp]=1;
f[j]=1;}
temp=temp+1;
}
quotient[i]=1;
}
else
quotient[i]=0;
}
printf("\nthe quotient is \n");
for(i=0;i<8;i++)
printf("%d",quotient[i]);
printf("\n and the remainder is \n ");
for(j=0;j<4;j++)
printf("%d",f[j]);
}
```

OUTPUT


root@Desktop-11:~/Desktop
File Edit View Search Terminal Help
[root@Desktop-11 Desktop]# ./a.out
enter the number
1 0 1 0 0 0 0 1
enter the divisor
1 0 0 1
the quotient is
10110111
and the remainder is
0111[root@Desktop-11 Desktop]#
[root@Desktop-11 Desktop]#
[root@Desktop-11 Desktop]#
[root@Desktop-11 Desktop]#
[root@Desktop-11 Desktop]#
[root@Desktop-11 Desktop]#
[root@Desktop-11 Desktop]#
[root@Desktop-11 Desktop]#
[root@Desktop-11 Desktop]#
[root@Desktop-11 Desktop]#
[root@Desktop-11 Desktop]#
[root@Desktop-11 Desktop]#
[root@Desktop-11 Desktop]#
[root@Desktop-11 Desktop]#