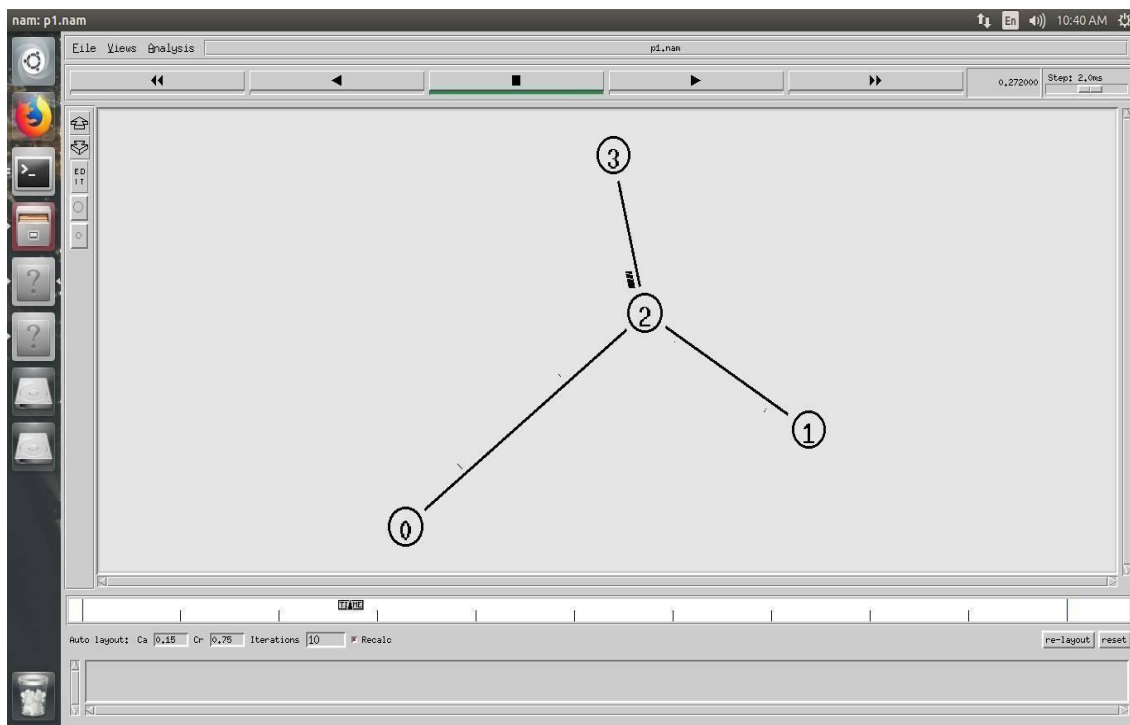
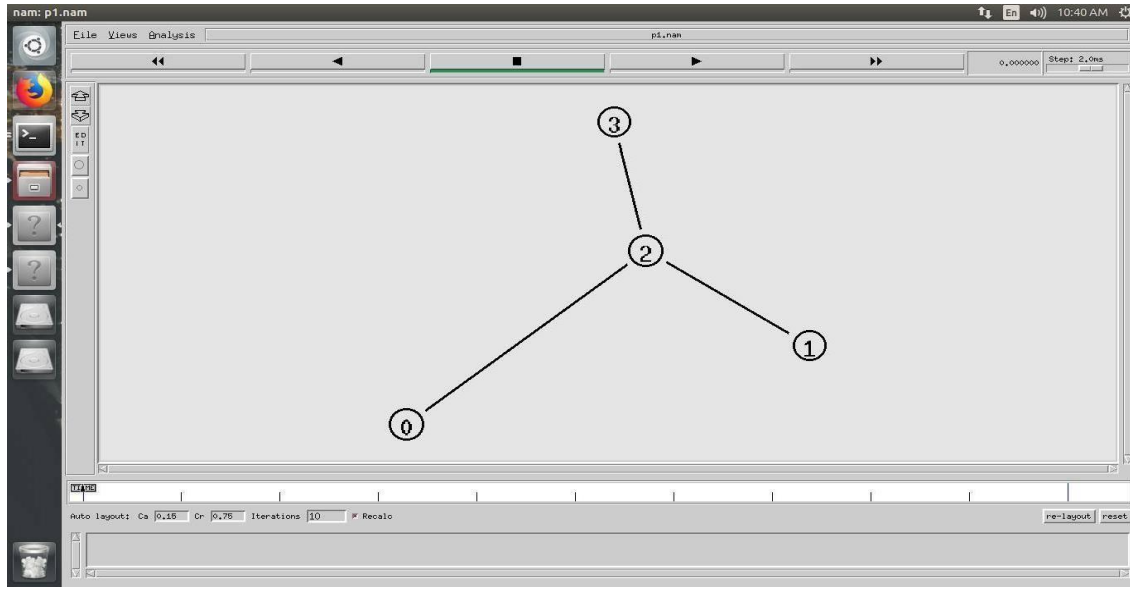
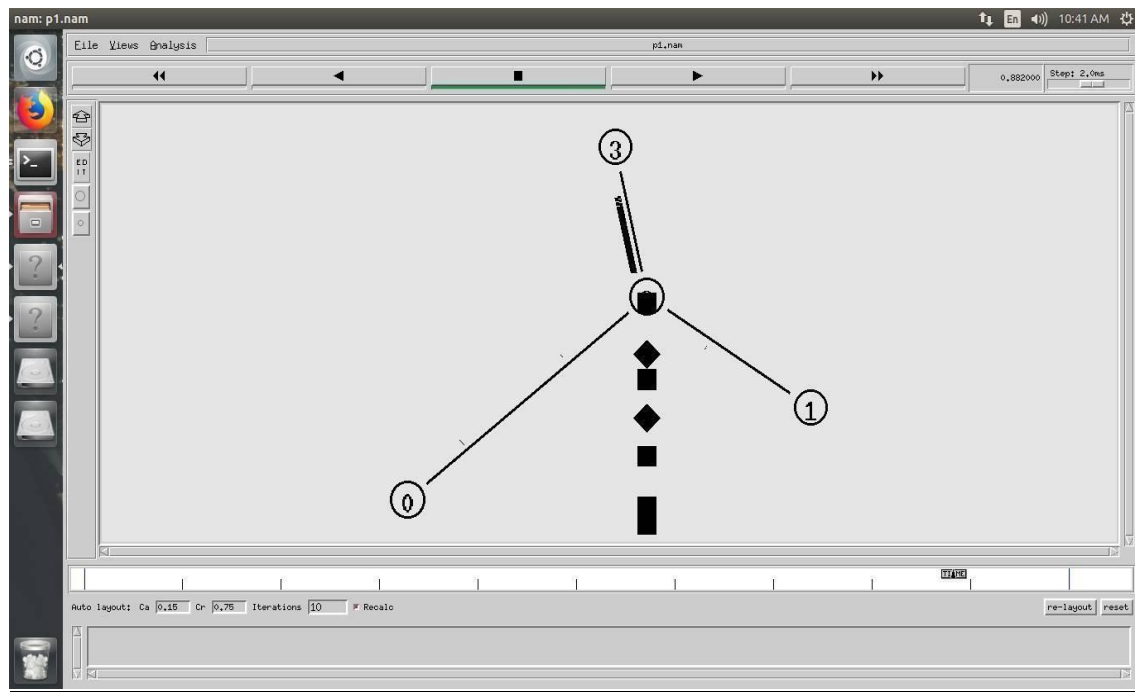


Trace file contains 12 columns:-

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

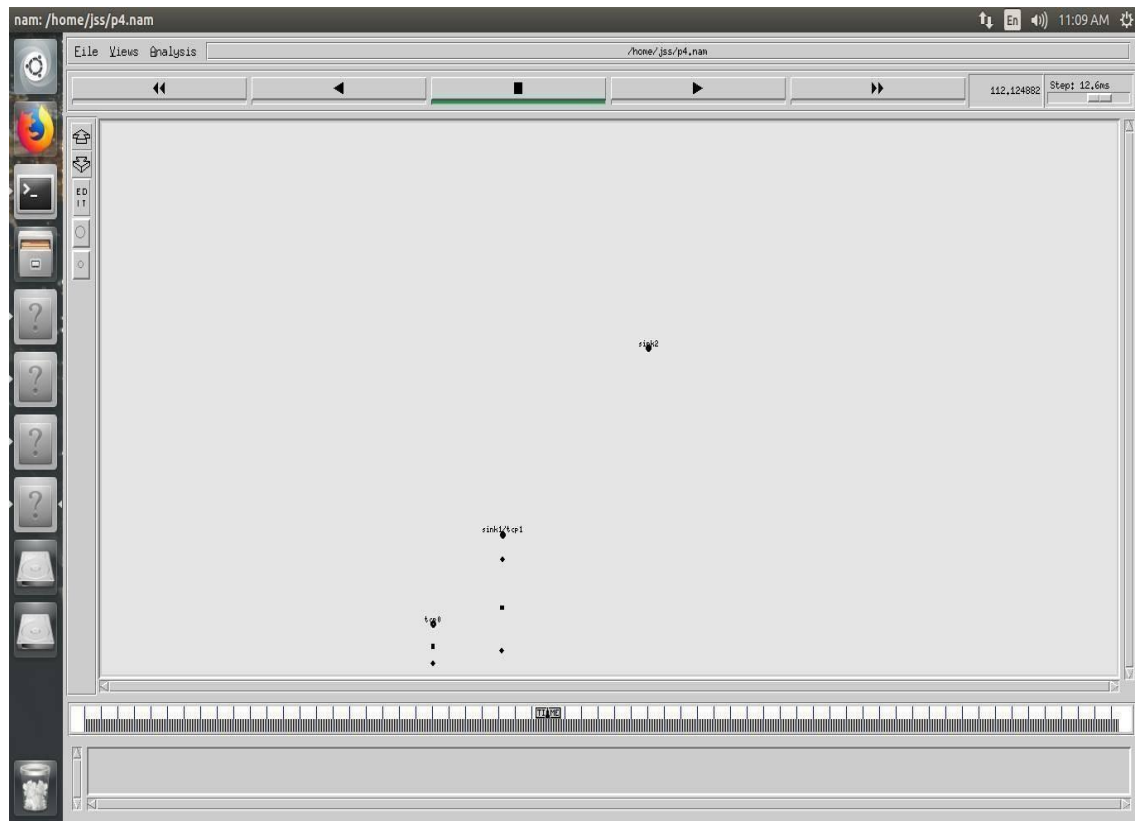
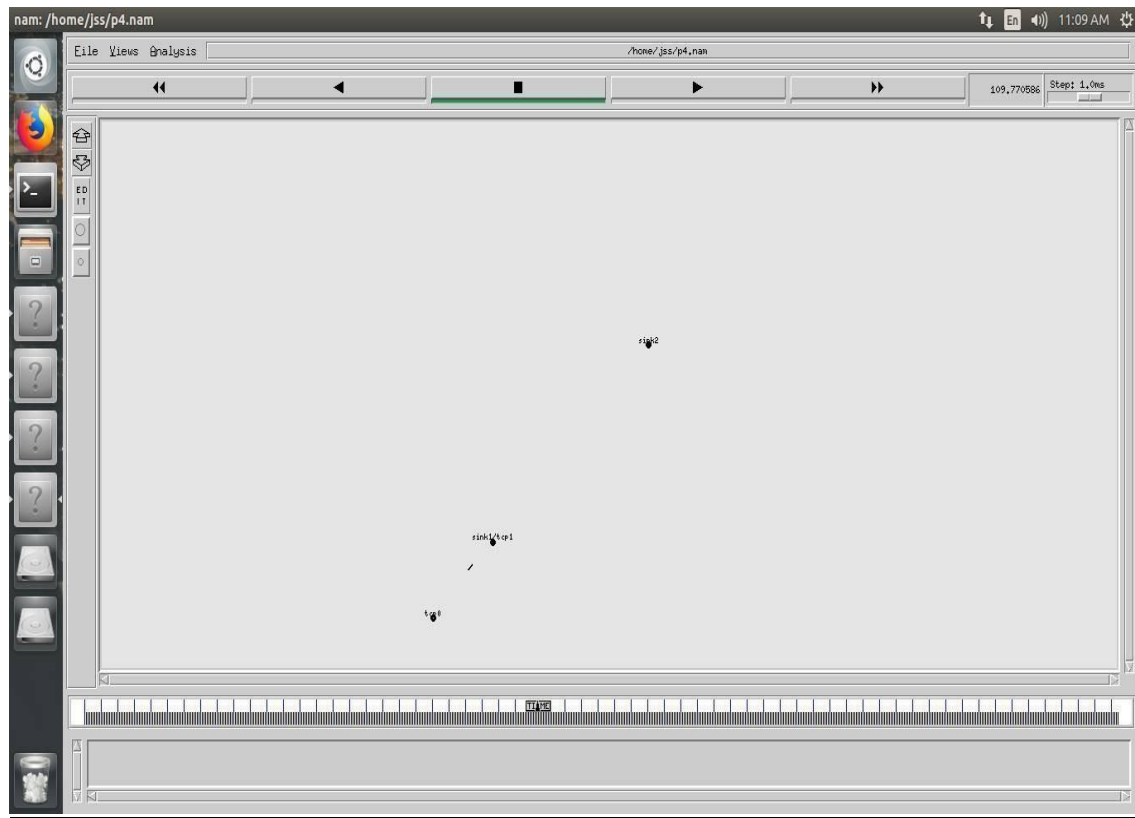


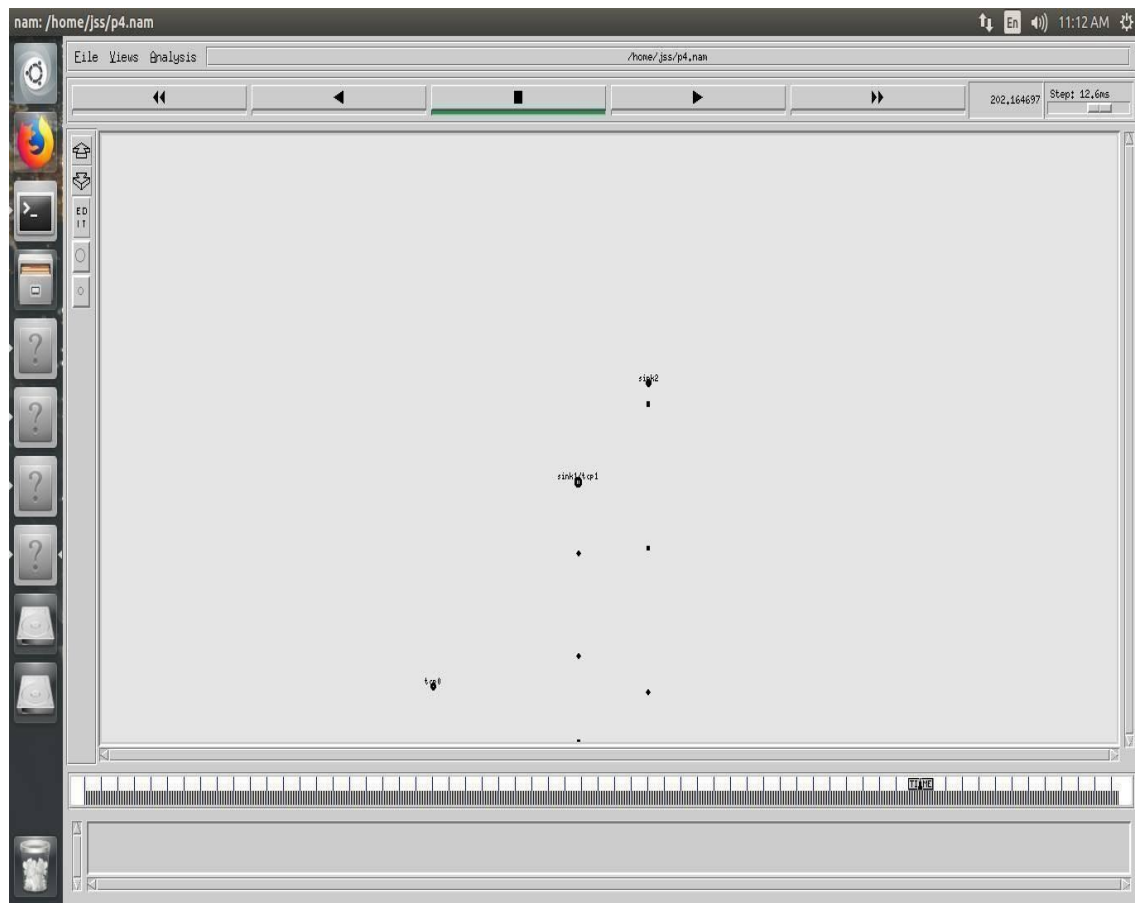
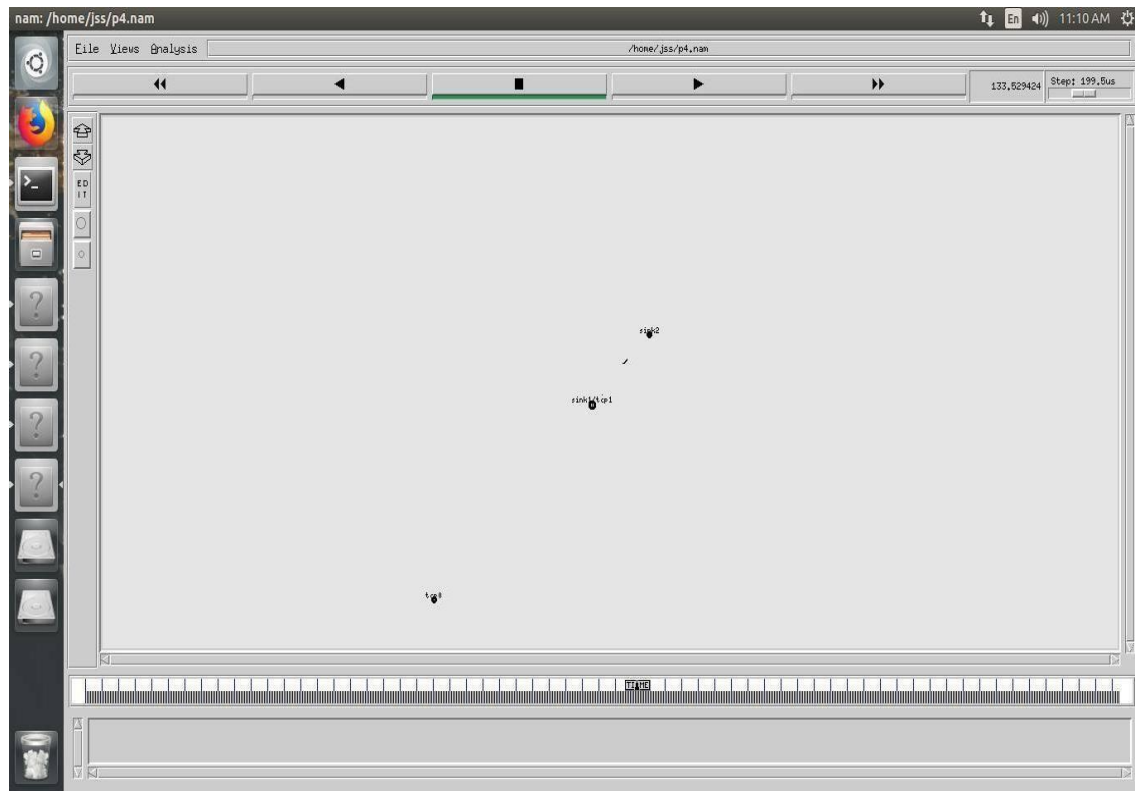
Output

```
jss@jss-OptiPlex-3046: ~
jss@jss-OptiPlex-3046:~$ awk -f p1.awk p1.tr
cbr      139
cbr      126
cbr      127
cbr      130
cbr      151
cbr      154
cbr      136
cbr      159
cbr      141
cbr      142
cbr      145
cbr      171
cbr      174
cbr      151
cbr      154
cbr      157
cbr      187
cbr      161
cbr      163
cbr      195
cbr      201
cbr      173
cbr      175
cbr      209
the no of packets dropped=24
jss@jss-OptiPlex-3046:~$
```

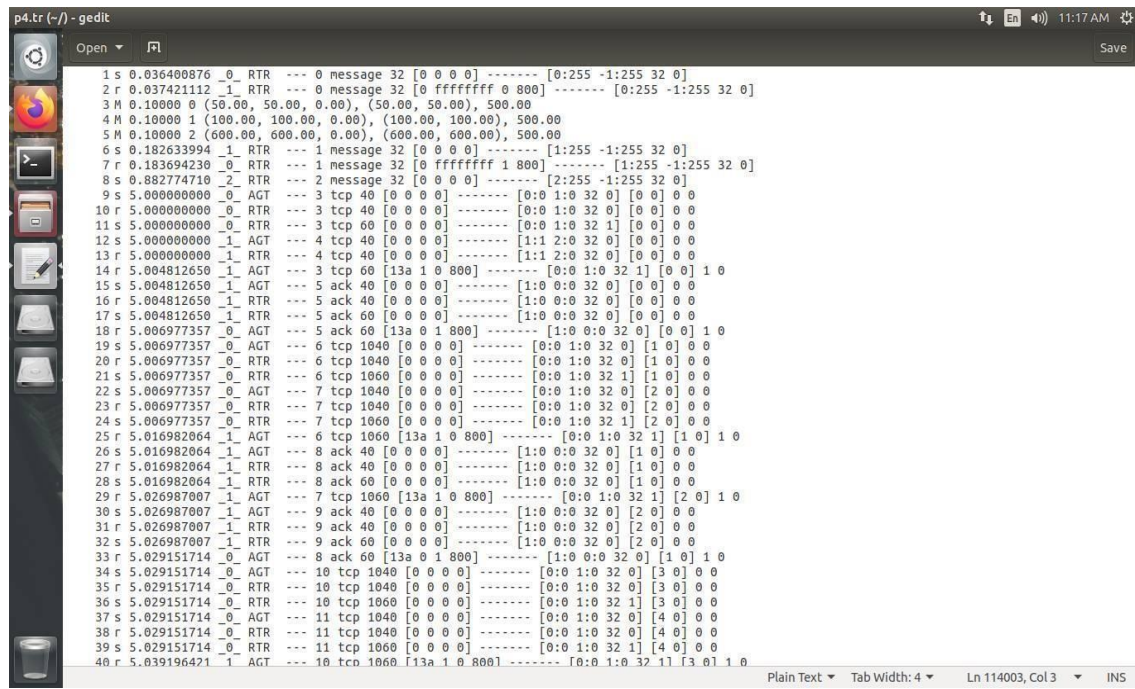
Note:

1. Set the queue size fixed from n_0 to n_2 as 10, n_1 - n_2 to 10 and from n_2 - n_3 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

Topology



Trace file



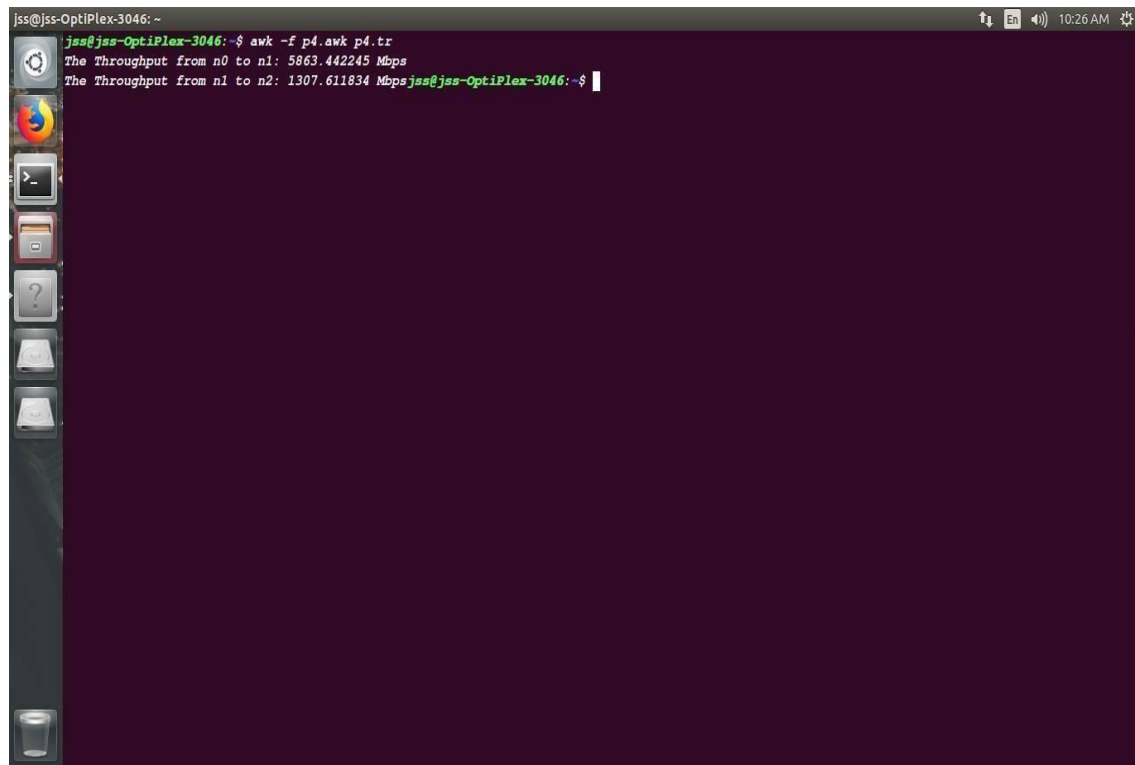
```

p4.tr (-) - gedit
1 s 0.036400870 _0 RTR --- 0 message 32 [0 0 0 0] ----- [0:255 -1:255 32 0]
2 r 0.037421112 _1 RTR --- 0 message 32 [0 ffffffff 0 800] ----- [0:255 -1:255 32 0]
3 M 0.10000 0 (50.00, 50.00, 0.00), (50.00, 50.00), 500.00
4 M 0.10000 1 (100.00, 100.00, 0.00), (100.00, 100.00), 500.00
5 M 0.10000 2 (600.00, 600.00, 0.00), (600.00, 600.00), 500.00
6 s 0.182633994 _1 RTR --- 1 message 32 [0 0 0 0] ----- [1:255 -1:255 32 0]
7 r 0.183694238 _0 RTR --- 1 message 32 [0 ffffffff 1 800] ----- [1:255 -1:255 32 0]
8 s 0.882774710 _2 RTR --- 2 message 32 [0 0 0 0] ----- [2:255 -1:255 32 0]
9 s 5.000000000 _0 AGT --- 3 tcp 40 [0 0 0 0] ----- [0:0 1:0 32 0] [0 0] 0 0
10 r 5.000000000 _0 RTR --- 3 tcp 40 [0 0 0 0] ----- [0:0 1:0 32 0] [0 0] 0 0
11 s 5.000000000 _0 RTR --- 3 tcp 60 [0 0 0 0] ----- [0:0 1:0 32 1] [0 0] 0 0
12 s 5.000000000 _1 AGT --- 4 tcp 40 [0 0 0 0] ----- [1:1 2:0 32 0] [0 0] 0 0
13 r 5.000000000 _1 RTR --- 4 tcp 40 [0 0 0 0] ----- [1:1 2:0 32 0] [0 0] 0 0
14 r 5.004812650 _1 AGT --- 3 tcp 60 [13a 1 0 800] ----- [0:0 1:0 32 1] [0 0] 1 0
15 s 5.004812650 _1 AGT --- 5 ack 40 [0 0 0 0] ----- [1:0 0:0 32 0] [0 0] 0 0
16 r 5.004812650 _1 RTR --- 5 ack 40 [0 0 0 0] ----- [1:0 0:0 32 0] [0 0] 0 0
17 s 5.004812650 _1 RTR --- 5 ack 60 [0 0 0 0] ----- [1:0 0:0 32 0] [0 0] 0 0
18 r 5.006977357 _0 AGT --- 5 ack 60 [13a 0 1 800] ----- [1:0 0:0 32 0] [0 0] 1 0
19 s 5.006977357 _0 AGT --- 6 tcp 1040 [0 0 0 0] ----- [0:0 1:0 32 0] [1 0] 0 0
20 r 5.006977357 _0 RTR --- 6 tcp 1040 [0 0 0 0] ----- [0:0 1:0 32 0] [1 0] 0 0
21 s 5.006977357 _0 RTR --- 6 tcp 1060 [0 0 0 0] ----- [0:0 1:0 32 1] [1 0] 0 0
22 s 5.006977357 _0 AGT --- 7 tcp 1040 [0 0 0 0] ----- [0:0 1:0 32 0] [2 0] 0 0
23 r 5.006977357 _0 RTR --- 7 tcp 1040 [0 0 0 0] ----- [0:0 1:0 32 0] [2 0] 0 0
24 s 5.006977357 _0 RTR --- 7 tcp 1060 [0 0 0 0] ----- [0:0 1:0 32 1] [2 0] 0 0
25 r 5.016982064 _1 AGT --- 6 tcp 1060 [13a 1 0 800] ----- [0:0 1:0 32 1] [1 0] 1 0
26 s 5.016982064 _1 AGT --- 8 ack 40 [0 0 0 0] ----- [1:0 0:0 32 0] [1 0] 0 0
27 r 5.016982064 _1 RTR --- 8 ack 40 [0 0 0 0] ----- [1:0 0:0 32 0] [1 0] 0 0
28 s 5.016982064 _1 RTR --- 8 ack 60 [0 0 0 0] ----- [1:0 0:0 32 0] [1 0] 0 0
29 r 5.026987007 _1 AGT --- 7 tcp 1060 [13a 1 0 800] ----- [0:0 1:0 32 1] [2 0] 1 0
30 s 5.026987007 _1 AGT --- 9 ack 40 [0 0 0 0] ----- [1:0 0:0 32 0] [2 0] 0 0
31 r 5.026987007 _1 RTR --- 9 ack 40 [0 0 0 0] ----- [1:0 0:0 32 0] [2 0] 0 0
32 s 5.026987007 _1 RTR --- 9 ack 60 [0 0 0 0] ----- [1:0 0:0 32 0] [2 0] 0 0
33 r 5.029151714 _0 AGT --- 8 ack 60 [13a 0 1 800] ----- [1:0 0:0 32 0] [1 0] 1 0
34 s 5.029151714 _0 AGT --- 10 tcp 1040 [0 0 0 0] ----- [0:0 1:0 32 0] [3 0] 0 0
35 r 5.029151714 _0 RTR --- 10 tcp 1040 [0 0 0 0] ----- [0:0 1:0 32 0] [3 0] 0 0
36 s 5.029151714 _0 RTR --- 10 tcp 1060 [0 0 0 0] ----- [0:0 1:0 32 1] [3 0] 0 0
37 s 5.029151714 _0 AGT --- 11 tcp 1040 [0 0 0 0] ----- [0:0 1:0 32 0] [4 0] 0 0
38 r 5.029151714 _0 RTR --- 11 tcp 1040 [0 0 0 0] ----- [0:0 1:0 32 0] [4 0] 0 0
39 s 5.029151714 _0 RTR --- 11 tcp 1060 [0 0 0 0] ----- [0:0 1:0 32 1] [4 0] 0 0
40 r 5.039196421 _1 AGT --- 10 tcp 1060 [13a 1 0 800] ----- [0:0 1:0 32 1] [3 0] 1 0

```

Here “M” indicates mobile nodes, “AGT” indicates Agent Trace, “RTR” indicates Router Trace

Output



```

jss@jss-OptiPlex-3046: ~
jss@jss-OptiPlex-3046:~$ awk -f p4.awk p4.tr
The Throughput from n0 to n1: 5863.442245 Mbps
The Throughput from n1 to n2: 1307.611834 Mbpsjss@jss-OptiPlex-3046:~$

```

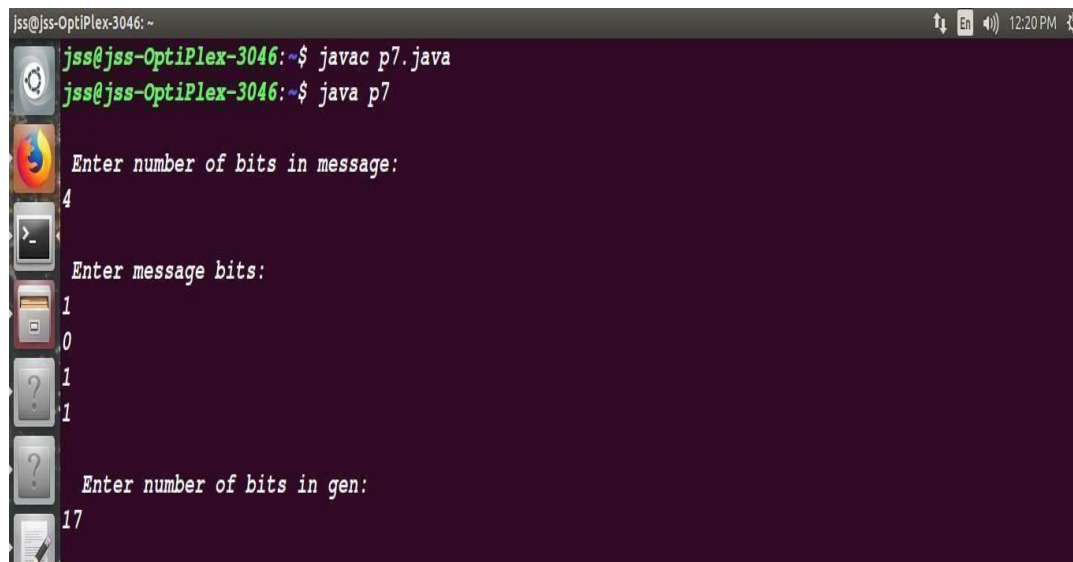
```
{
int current=0;
while(true)
{
for(int i=0;i<gen.length;i++)
{
rem[current+i]=(rem[current+i]^gen[i]);
}
while(rem[current]==0 && current!=rem.length-1)
{
current++;
}
if((rem.length-current)<gen.length)
{
break;
}
}
return rem;
}
}
```

Output:

jss@jss-Optiplex-3046:~\$ vi p7.java

jss@jss-Optiplex-3046:~\$ javac p7.java

jss@jss-Optiplex-3046:~\$ java p7



```
jss@jss-OptiPlex-3046: ~
jss@jss-OptiPlex-3046:~$ javac p7.java
jss@jss-OptiPlex-3046:~$ java p7
Enter number of bits in message:
4
Enter message bits:
1
0
1
1
Enter number of bits in gen:
17
```

A screenshot of a Linux terminal window titled "jss@jss-OptiPlex-3046: ~". The terminal displays a sequence of bits (0s and 1s) being entered one by one. A vertical dock on the left contains icons for various applications like Firefox, LibreOffice, and a file manager. At the bottom, there are system status icons for network, language (En), and volume, along with the time "12:20 PM".

```
jss@jss-OptiPlex-3046: ~  
  
Enter gen bits :  
1  
0  
0  
0  
1  
0  
0  
0  
0  
0  
0  
0  
1  
0  
0  
0  
0  
1  
  
Message bits are:1011  
Generators bits are:100010000000100001  
Appended message is:10110000000000000000*  
Transmitted message from the transmitteris :  
101110110001011101011  
Enter received message of +total_bits+ at receiver end:
```

```
jss@jss-OptiPlex-3046: ~  
Message bits are:1011  
Generators bits are:10001000000100001  
Appended message is:10110000000000000000*  
Transmitted message from the transmitter is :  
101110110001011101011  
>_ Enter received message of +total_bits+ at receiver end:  
1  
0  
1  
1  
1  
0  
1  
0  
1  
0  
1  
0  
1  
0  
1  
0  
1  
0  
1  
1  
1
```

[illegible]

```
jss@jss-OptiPlex-3046: ~  
jss@jss-OptiPlex-3046: ~$ javac p7.java  
jss@jss-OptiPlex-3046: ~$ java p7  
Enter number of bits in message:  
4  
Enter message bits:  
1  
0  
1  
1  
Enter number of bits in gen:  
17
```


A screenshot of a Linux terminal window titled "jss@jss-OptiPlex-3046: ~". The terminal displays the output of a C program that implements CRC error detection. On the left side of the terminal, there is a vertical dock containing icons for various applications: a gear (system settings), Firefox, a file manager, a terminal, a calendar, a help icon (?), another help icon (?), a document with a pencil, a clock, a folder, and a trash can. The program's output is as follows:

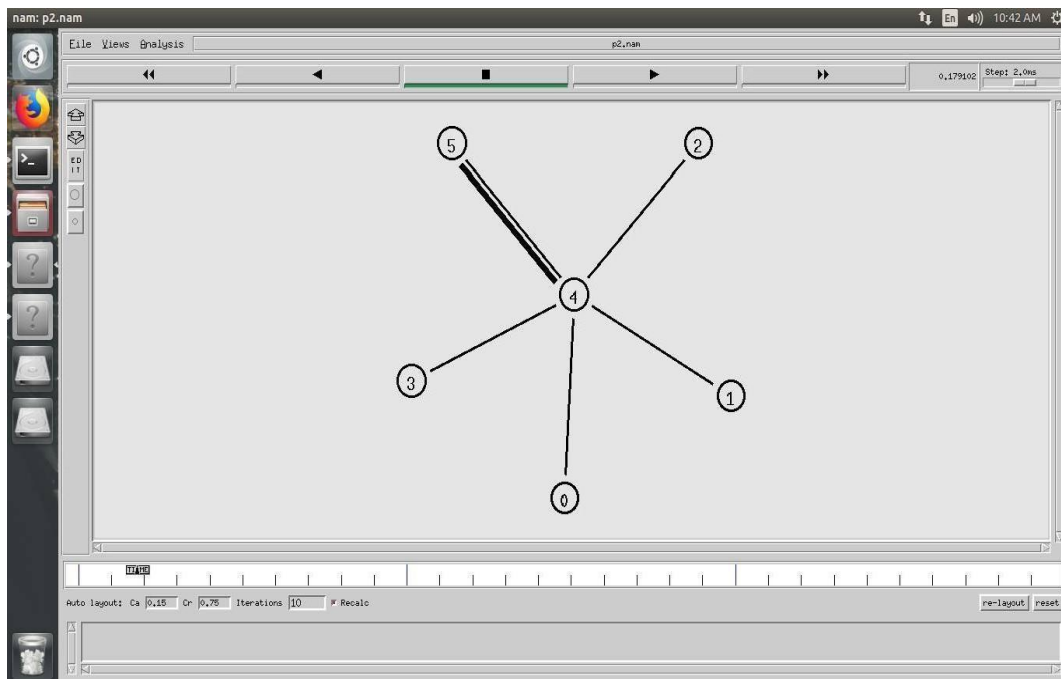
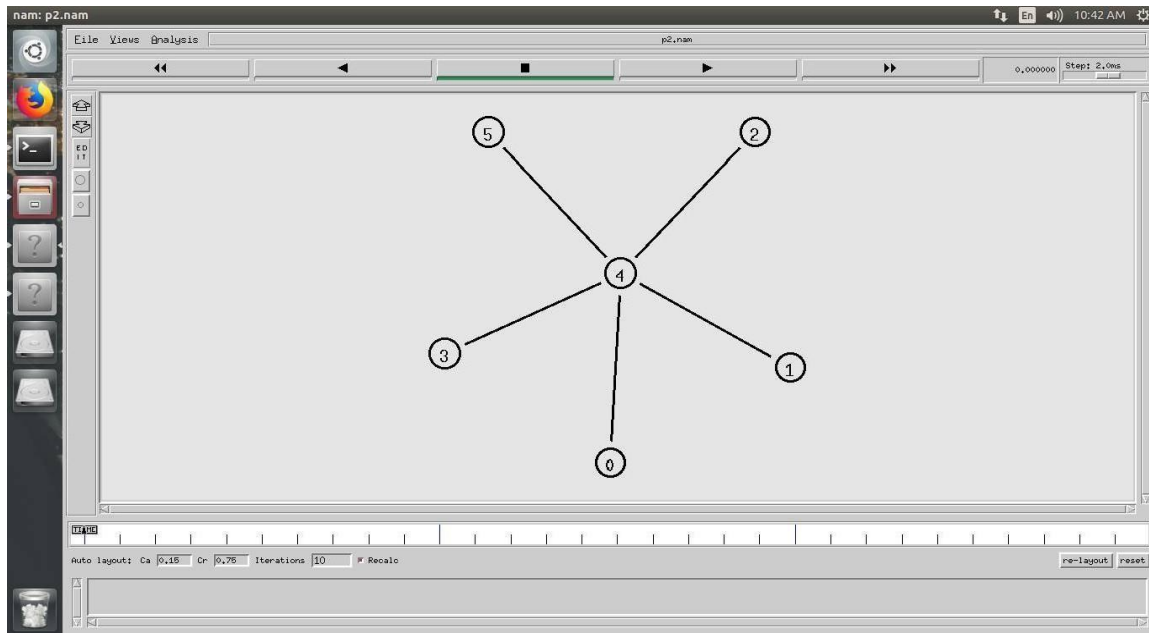
```
Enter gen bits :  
1  
0  
0  
0  
1  
0  
0  
0  
0  
0  
0  
1  
0  
0  
0  
1  
  
Message bits are:1011  
Generators bits are:10001000000100001  
Appended message is:10110000000000000000*  
Transmitted message from the transmitteris :  
101110110001011101011  
Enter received message of +total_bits+ at receiver end:
```

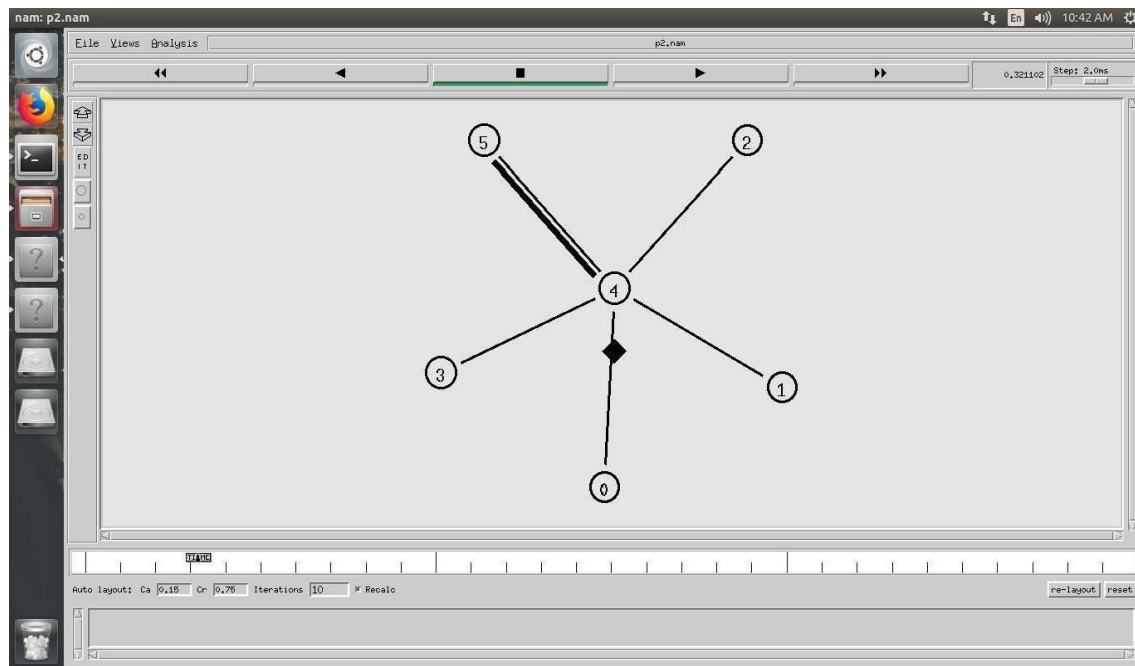
The program prompts for generator bits, which are entered as 10001000000100001. It then calculates the appended message (10110000000000000000*) and the transmitted message (101110110001011101011). Finally, it prompts for the received message.

[illegible]

- 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 begin.
- 6) After simulation is completed run **awk file** to see the output ,
[root@localhost~]# **awk -f p2.awk p2.tr**
- 7) To see the trace file contents open the file as ,
[root@localhost~]# **vi p2.tr**

Topology





```

jss@jss-OptiPlex-3046: ~$ ns p2.tcl
node 2received answer from 3 with round trip time 5.3 msec
node 2received answer from 3 with round trip time 5.3 msec
node 2received answer from 3 with round trip time 5.3 msec
node 2received answer from 3 with round trip time 5.3 msec
node 0received answer from 5 with round trip time 404.9 msec
node 2received answer from 3 with round trip time 5.3 msec
node 2received answer from 3 with round trip time 5.3 msec
node 2received answer from 3 with round trip time 5.3 msec
node 2received answer from 3 with round trip time 5.3 msec
node 0received answer from 5 with round trip time 704.9 msec
node 2received answer from 3 with round trip time 5.3 msec
node 2received answer from 3 with round trip time 5.3 msec
node 2received answer from 3 with round trip time 5.3 msec
node 2received answer from 3 with round trip time 5.3 msec
node 0received answer from 5 with round trip time 804.9 msec
node 2received answer from 3 with round trip time 5.3 msec
node 2received answer from 3 with round trip time 5.3 msec
node 2received answer from 3 with round trip time 5.3 msec
node 2received answer from 3 with round trip time 5.3 msec
node 0received answer from 5 with round trip time 804.9 msec
node 2received answer from 3 with round trip time 5.3 msec
node 2received answer from 3 with round trip time 5.3 msec
node 2received answer from 3 with round trip time 5.3 msec
node 2received answer from 3 with round trip time 5.3 msec
node 0received answer from 5 with round trip time 804.9 msec
  
```

Output

```

jss@jss-OptiPlex-3046: ~$ awk -f p2.awk p2.tr
Total number of ping packets dropped due to congestion =20
jss@jss-OptiPlex-3046: ~$
  
```

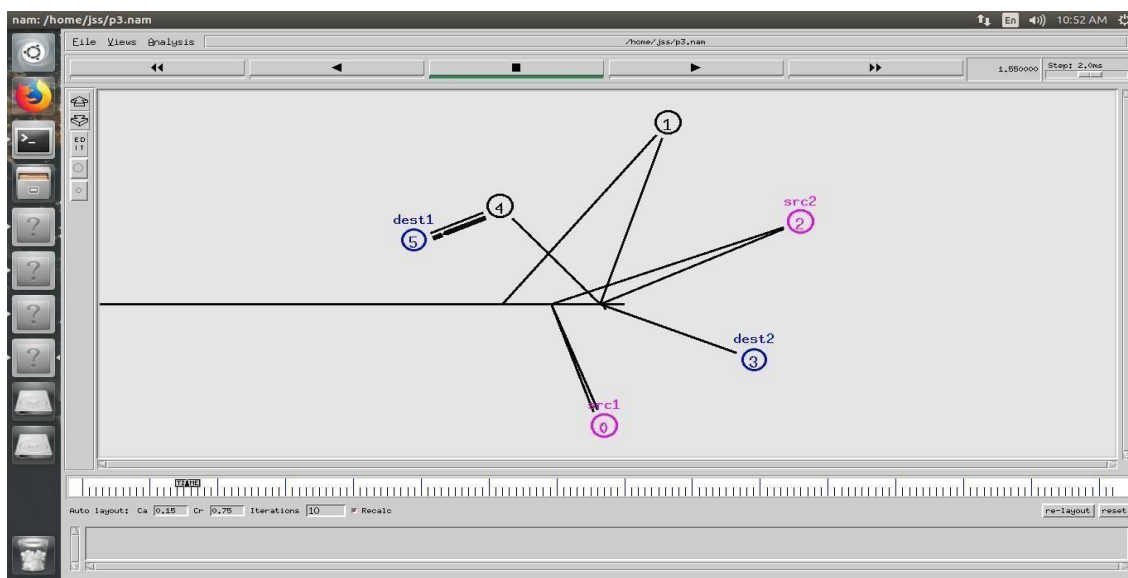
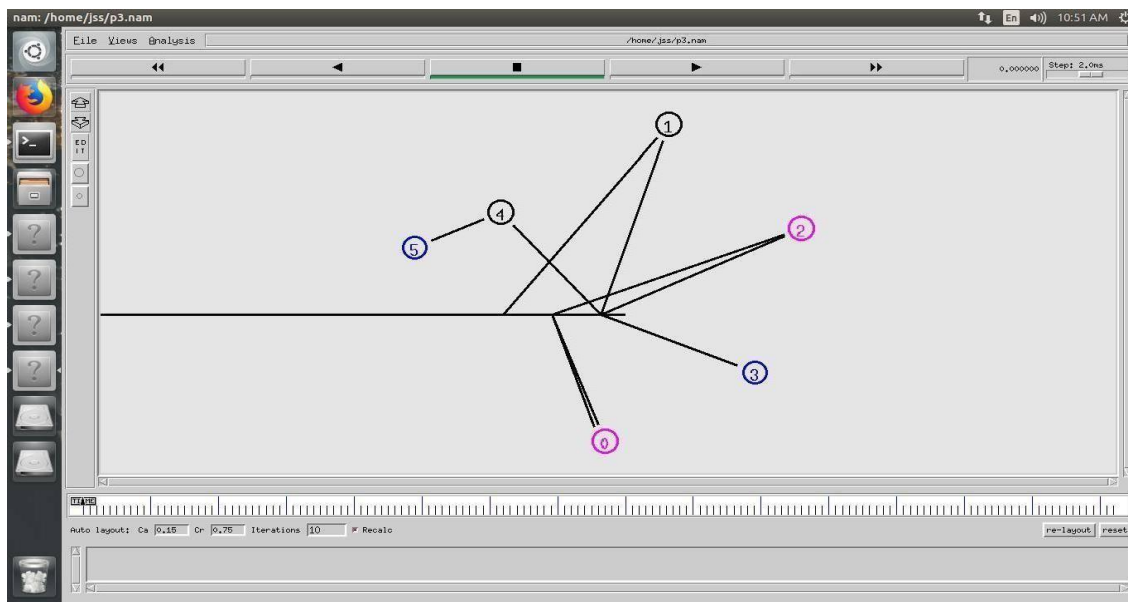
Note:

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.

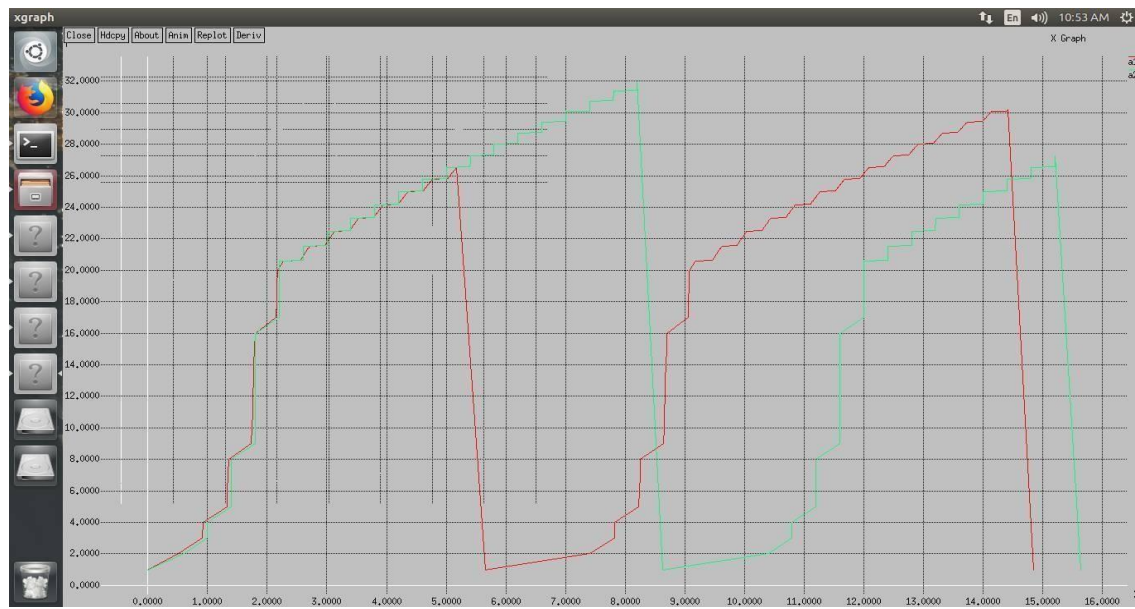
```
jss@jss-OptiPlex-3046: ~  
jss@jss-OptiPlex-3046:~$ javac p8.java  
jss@jss-OptiPlex-3046:~$ java p8  
enter the num of vertices  
4  
enter the adjacency matrix:  
0 5 0 0  
5 0 3 4  
0 3 0 2  
0 4 2 0  
enter the source vertex  
2  
distance of source 2 to 1 is 5  
distance of source 2 to 2 is 0  
distance of source 2 to 3 is 3  
distance of source 2 to 4 is 4
```

- 5) Run the simulation program
`[root@localhost~]# ns p3.tcl`
- 6) After simulation is completed run **awk file** to see the output ,
 - i. `[root@localhost~]# awk -f p3.awk file1.tr > a1`
 - ii. `[root@localhost~]# awk -f p3.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 p3.tr`

Topology



Output



Output:

```
jss@jss-OptiPlex-3046: ~$ javac p12.java
jss@jss-OptiPlex-3046: ~$ java p12
enter the bs,outgoing rate,inputs,incoming size
7
5
10
8
incoming size is 8
packet lost = 1
bucket buffersize is 7 out of 7
after outgoing = 2 packet left out of 7 in buffer
incoming size is 8
packet lost = 3
bucket buffersize is 7 out of 7
after outgoing = 2 packet left out of 7 in buffer
incoming size is 8
packet lost = 3
bucket buffersize is 7 out of 7
after outgoing = 2 packet left out of 7 in buffer
incoming size is 8
packet lost = 3
bucket buffersize is 7 out of 7
after outgoing = 2 packet left out of 7 in buffer
incoming size is 8
packet lost = 3
bucket buffersize is 7 out of 7
after outgoing = 2 packet left out of 7 in buffer
```

[illegible]