**SIR M VISVESVARAYA INSTITUTE OF TECHNOLOGY**

*(Affiliated to VTU, Recognized by AICTE and Accredited by NBA, NAAC
and an ISO 9001-2008 Certified Institution)*
Bengaluru – 562157

# DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

# COMPUTER NETRWORKS LABORATORY

**CHOICE BASEDCREDITSYSTEM**

**21CS52- V Semester B.E**
**(*Academic Year 2023-24*)**

**Compiled and Prepared by:**

**Ms. Savitha P, Mr. Elaiyaraja P**
**Assistant Professor**
**Dept. of CSE**

**Under the Guidance of:**

**Dr. Anitha T N**
**Professor & Head**
**Dept. of CSE**

## Department Vision and Mission

### VISION

To build a center for imparting quality technical education and carrying out research activity to meet the current and future challenges in the domain of Computer Science and Engineering.

### MISSION

- The Computer Science and Engineering department strives for excellence in teaching, applying, promoting and imparting knowledge through comprehensive academic curricula.
- Train students to effectively apply the knowledge to solve real-world problems, thus enhance their potential for life-long high-quality career and give them a competitive advantage in the ever-changing and fast paced computing.
- Prepare students to demonstrate a sense of societal and ethical responsibilities in their professional endeavors.
- Creating amongst students and faculty a collaborative environment open to the free exchange of ideas, which leads to research activity and fuels innovation thinking.

## PROGRAM OUTCOMES

| PO's | PO Description |
|---|---|
| PO1 | **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems. |
| PO2 | **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences. |
| PO3 | **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations. |
| PO4 | **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions. |
| PO5 | **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations. |
| PO6 | **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice. |
| PO7 | **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development. |
| PO8 | **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice. |
| PO9 | **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings. |
| PO10 | **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions. |
| PO11 | **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments. |
| PO12 | **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change. |

## PROGRAM SPECIFIC OUTCOMES

| PSO's | PSO Description |
|---|---|
| PSO1 | An ability to design and analyze algorithms by applying theoretical concepts to build complex and computer- based systems in the domain of System Software, Computer Networks & Security, Web technologies, Data Science and Analytics. |
| PSO2 | Be able to develop various software solutions by applying the techniques of Data Base Management, Complex Mathematical Models, Software Engineering practices and Machine Learning with Artificial Intelligence. |

# COMPUTER NETWORKS

| | | | |
|---|---|---|---|
| Course Code: | 21CS52 | CIE Marks | 50 |
| Teaching Hours/Week (L:T:P: S) | 3:0:2:0 | SEE Marks | 50 |
| Total Hours of Pedagogy | 40T + 20P | Total Marks | 100 |
| Credits | 04 | Exam Hours | 03 |

**Course Objectives:**

CLO 1. Fundamentals of data communication networks.
CLO 2. Software and hardware interfaces
CLO 3. Application of various physical components and protocols
CLO 4. Communication challenges and remedies in the networks.

**Teaching-Learning Process (General Instructions)**

These are sample Strategies, which teachers can use to accelerate the attainment of the various course outcomes.

1. Lecturer method (L) need not to be only traditional lecture method, but alternative effective teaching methods could be adopted to attain the outcomes.
2. Use of Video/Animation to explain functioning of various concepts.
3. Encourage collaborative (Group Learning) Learning in the class.
4. Ask at least three HOT (Higher order Thinking) questions in the class, which promotes criticalthinking.
5. Adopt Problem Based Learning (PBL), which fosters students' Analytical skills, develop designthinking skills such as the ability to design, evaluate, generalize, and analyze information rather than simply recall it.
6. Introduce Topics in manifold representations.
7. Show the different ways to solve the same problem and encourage the students to come upwith their own creative ways to solve them.
8. Discuss how every concept can be applied to the real world - and when that's possible, it helps improve the students' understanding.

## Module-1

**Introduction to networks**: Network hardware, Network software, Reference models,
**Physical Layer:** Guided transmission media, Wireless transmission
**Textbook 1: Ch.1.2 to 1.4, Ch.2.2 to 2.3**

*Laboratory Component:*

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

| **Teaching-Learning Process** | Chalk and board, Problem based learning, Demonstration |
|---|---|

## Module-2

**The Data link layer:** Design issues of DLL, Error detection and correction, Elementary data linkprotocols, Sliding window protocols.
**The medium access control sublayer:** The channel allocation problem, Multiple access protocols.
**Textbook 1: Ch.3.1 to 3.4, Ch.4.1 and 4.2**

*Laboratory Component:*

1. Implement simple ESS and with transmitting nodes in wire-less LAN by simulation anddetermine the throughput with respect to transmission of packets
2. Write a program for error detecting code using CRC-CCITT (16- bits).

| **Teaching-Learning Process** | Chalk and board, Problem based learning, Demonstration |
|---|---|

## Module-3

**The Network Layer:**
Network Layer Design Issues, Routing Algorithms, Congestion Control Algorithms, QoS.
**Textbook 1: Ch 5.1 to 5.4**

*Laboratory Component:*

1. Implement transmission of ping messages/trace route over a network topology consisting of 6nodes and find the number of packets dropped due to congestion in the network.
2. Write a program to find the shortest path between vertices using bellman-ford algorithm.

| **Teaching-Learning Process** | Chalk and board, Problem based learning, Demonstration |
|---|---|

## Module-4

**The Transport Layer:** The Transport Service, Elements of transport protocols, Congestion control, Theinternet transport protocols.
**Textbook 1: Ch 6.1 to 6.4 and 6.5.1 to 6.5.7**

***Laboratory Component:***

1. Implement an Ethernet LAN using n nodes and set multiple traffic nodes and plot congestionwindow for different source / destination.
2. Write a program for congestion control using leaky bucket algorithm.

| | |
|---|---|
| **Teaching-Learning Process** | Chalk and board, Problem based learning, Demonstration |

**Module-5**

**Application Layer:** Principles of Network Applications, The Web and HTTP, Electronic Mail in the Internet, DNS—The Internet's Directory Service.

**Textbook 2: Ch 2.1 to 2.4**

| | |
|---|---|
| **Teaching-Learning Process** | Chalk and board, Problem based learning, Demonstration |

**Course Outcomes (Course Skill Set)**

At the end of the course the student will be able to:

CO 1. Learn the basic needs of communication system.
CO 2. Interpret the communication challenges and its solution.
CO 3. Identify and organize the communication system network componentsCO 4. Design communication networks for user requirements.

**Assessment Details (both CIE and SEE)**

The weightage of Continuous Internal Evaluation (CIE) is 50% and for Semester End Exam (SEE) is 50%. The minimum passing mark for the CIE is 40% of the maximum marks (20 marks). A student shall be deemed to have satisfied the academic requirements and earned the credits allotted to each subject/ course if the student secures not less than 35% (18 Marks out of 50) in the semester-end examination (SEE), and a minimum of 40% (40 marks out of 100) in the sum total of the CIE (Continuous Internal Evaluation) and SEE (Semester End Examination) taken together

**Continuous Internal Evaluation:**

Three Unit Tests each of **20 Marks (duration 01 hour**)

1. First test at the end of 5th week of the semester
2. Second test at the end of the 10th week of the semester
3. Third test at the end of the 15th week of the semesterTwo

assignments each of **10 Marks**

4. First assignment at the end of 4th week of the semester
5. Second assignment at the end of 9th week of the semester

Practical Sessions need to be assessed by appropriate rubrics and viva-voce method. This will contribute to **20 marks**.

- Rubrics for each Experiment taken average for all Lab components – 15 Marks.
- Viva-Voce– 5 Marks (more emphasized on demonstration topics)

The sum of three tests, two assignments, and practical sessions will be out of 100 marks and will be

**scaled down to 50 marks**

(to have a less stressed CIE, the portion of the syllabus should not be common /repeated for any of the methods of the CIE. Each method of CIE should have a different syllabus portion of the course).

**CIE methods /question paper has to be designed to attain the different levels of Bloom'staxonomy as per the outcome defined for the course.**

**Semester End Examination:**

Theory SEE will be conducted by University as per the scheduled timetable, with common questionpapers for the subject (**duration 03 hours)**

1. The question paper will have ten questions. Each question is set for 20 marks. Marks scoredshall be proportionally reduced to 50 marks
2. There will be 2 questions from each module. Each of the two questions under a module (with amaximum of 3 sub-questions), **should have a mix of topics** under that module.

The students have to answer 5 full questions, selecting one full question from each module

**Suggested Learning Resources:**

**Textbooks:**

1. Computer-Networks- Andrew S. Tanenbaum and David J. Wetherall, Pearson Education, 5th-Edition. (www.pearsonhighered.com/tanenbaum)
2. Computer Networking A Top-Down Approach -James F. Kurose and Keith W. RossPearsonEducation 7th Edition.

**Reference Books:**

1. Behrouz A Forouzan, Data and Communications and Networking, Fifth Edition, McGrawHill,Indian Edition
2. Larry L Peterson and Brusce S Davie, Computer Networks, fifth edition, ELSEVIER

**Weblinks and Video Lectures (e-Resources):**

1. https://www.digimat.in/nptel/courses/video/106105183/L01.html
2. http://www.digimat.in/nptel/courses/video/106105081/L25.html
3. https://nptel.ac.in/courses/106105081
4. VTU e-Shikshana Program

**Activity Based Learning (Suggested Activities in Class)/ Practical Based learning**

Simulation of Personal area network, Home area network, achieve QoS etc.

**Note**: For the Simulation experiments modify the topology and parameters set for the experiment and take multiple rounds of reading and analyze the results available in log files. Plot necessary graphs and conclude using NS2. Installation procedure of the required software must be demonstrated, carried outin groups, and documented in the report. Non simulation programs can be implemented using Java

The Sum of scaled-down marks scored in the report write-up/journal and average marks of two tests is the total CIE marks scored by the student.

**Semester End Evaluation (SEE):**

- SEE marks for the practical course is 50 Marks.
- SEE shall be conducted jointly by the two examiners of the same institute, examiners areappointed by the University
- All laboratory experiments are to be included for practical examination.
- (Rubrics) Breakup of marks and the instructions printed on the cover page of the answer script to be strictly adhered to by the examiners. **OR** based on the course requirement evaluation rubrics shall be decided jointly by examiners.
- Students can pick one question (experiment) from the questions lot prepared by the internal /external examiners jointly.
- Evaluation of test write-up/ conduction procedure and result/viva will be conducted jointly by examiners.
- General rubrics suggested for SEE are mentioned here, writeup-20%, Conduction procedure and result in -60%, Viva-voce 20% of maximum marks. SEE for practical shall be evaluated for 100 marks and scored marks shall be scaled down to 50 marks (however, based on course type, rubrics shall be decided by the examiners)
- *Students can pick one experiment from the questions lot of PART A with an equal choice to all the students in a batch. For PART B, the project group (Maximum of 4 students per batch) should demonstrate the mini-project.*
- *Weightage of marks for PART A is 60% and for PART B is 40%. General rubrics suggested to be followed for part A and part B.*
- Change of experiment is allowed only once and Marks allotted to the procedure part to be made zero (Not allowed for Part B).
- The duration of SEE is 03 hours

Rubrics suggested in Annexure-II of Regulation book

**Textbooks:**

1. Fundamentals of Database Systems, Ramez Elmasri and Shamkant B. Navathe, 7th Edition, 2017,Pearson.
2. Database management systems, Ramakrishnan, and Gehrke, 3rd Edition, 2014, McGraw Hill

**Suggested Weblinks/ E Resource**

https://www.tutorialspoint.com/sql/index.htm

# Introduction to Network Simulators

Network simulators implemented in software are valuable tools for researchers todevelop, test, and diagnose network protocols. Simulation is economical because it can carry out experiments without the actual hardware. It is flexible because it can, for example, simulate a link with any bandwidth and propagation delay. Simulation results are easier to analyze than experimental results because important information at critical points can be easily logged to help researchers diagnose network protocols.

Network simulators, however, have their limitations. A complete network simulator needs to simulate networking devices (e.g., hosts and routers) and application programs that generate network traffic. It also needs to provide network utility programs to configure, monitor, and gather statistics about a simulated network. Therefore, developing a complete network simulator is a large effort. Due to limited development resources, traditional network simulators usually have the following drawbacks:

• Simulation results are not as convincing as those produced by real hardware and software equipment. In order to constrain their complexity and development cost, most network simulators usually can only simulate real-life network protocol implementations with limited details, and this may lead to incorrect results.

• These simulators are not extensible in the sense that they lack the standard UNIX POSIX application programming interface (API). As such, existing or to-be-developed real-life application programs cannot run normally to generate traffic for a simulated network.

Instead, they must be rewritten to use the internal API provided by the simulator (if there is any) and be compiled with the simulator to form a single, big, and complex program.

To overcome these problems, Wang invented a kernel re-entering simulation methodology and used it to implement the Harvard network simulator. Later on, Wang further improved the methodology and used it to design and implement the NCTUns network simulator and emulator.

Different types of simulators

Some of the different types of simulators are as follows:-

- MIT's NETSIM
- NIST
- CPSIM
- INSANE
- NEST
- REAL
- NS
- OPNET
- NCTUns
- P
- GNS3
- WireShark

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

**Source Code :**

**pa1.tcl**

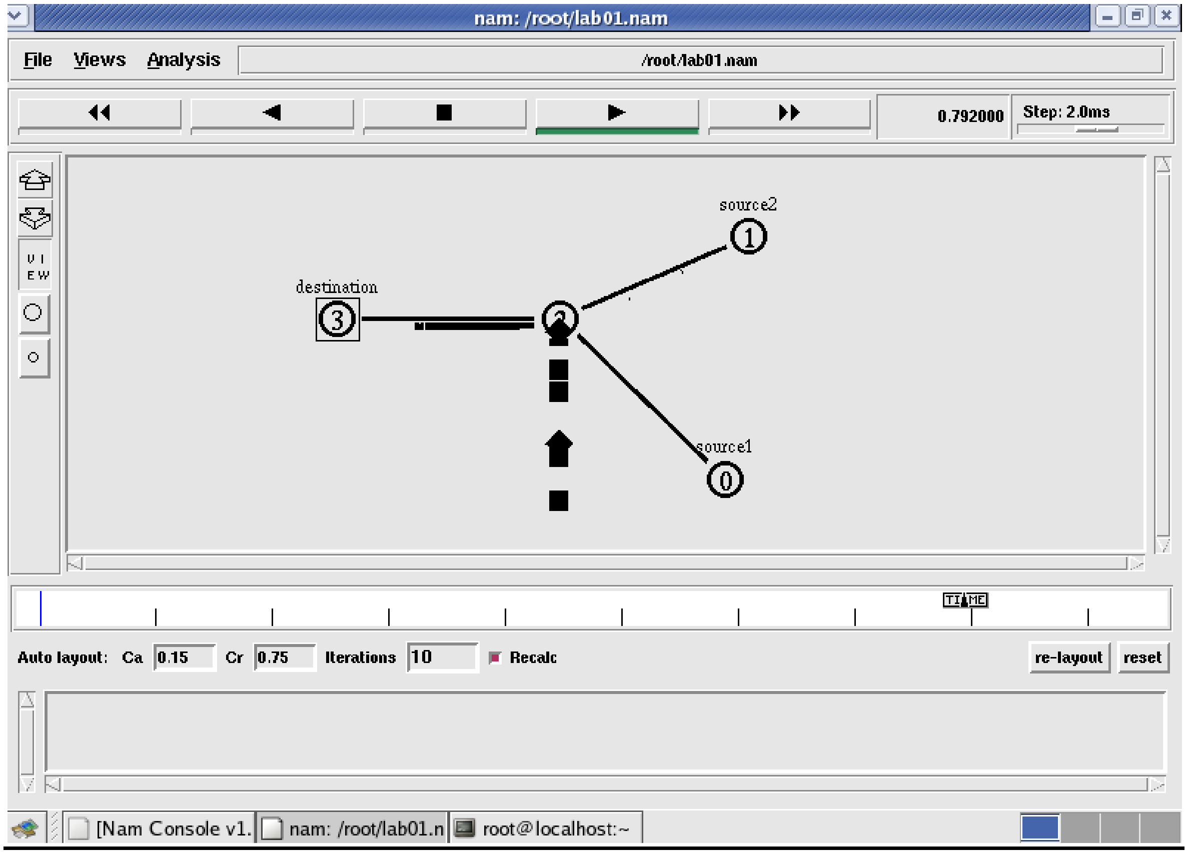
```
set ns [new Simulator]

set nf [open pa1.nam w]
$ns namtrace-all $nf

set tf [open pa1.tr w]
$ns trace-all $tf

proc finish {} {
global ns nf tf
$ns flush-trace
close $nf
close $tf
exec nam pa1.nam &
exit 0
}
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]

#$ns color 1 "red"
#$ns color 2 "blue"

$ns duplex-link $n0 $n2 20Mb 10ms DropTail
$ns duplex-link $n1 $n2 50Mb 5ms DropTail
$ns duplex-link $n2 $n3 10Mb 1000ms DropTail

#$ns duplex-link $n0 $n2 color "green"

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

set udp0 [new Agent/UDP]
$ns attach-agent $n0 $udp0

set cbr0 [new Application/Traffic/CBR]
```

```
$cbr0 set packetSize_ 500
$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]
$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
```

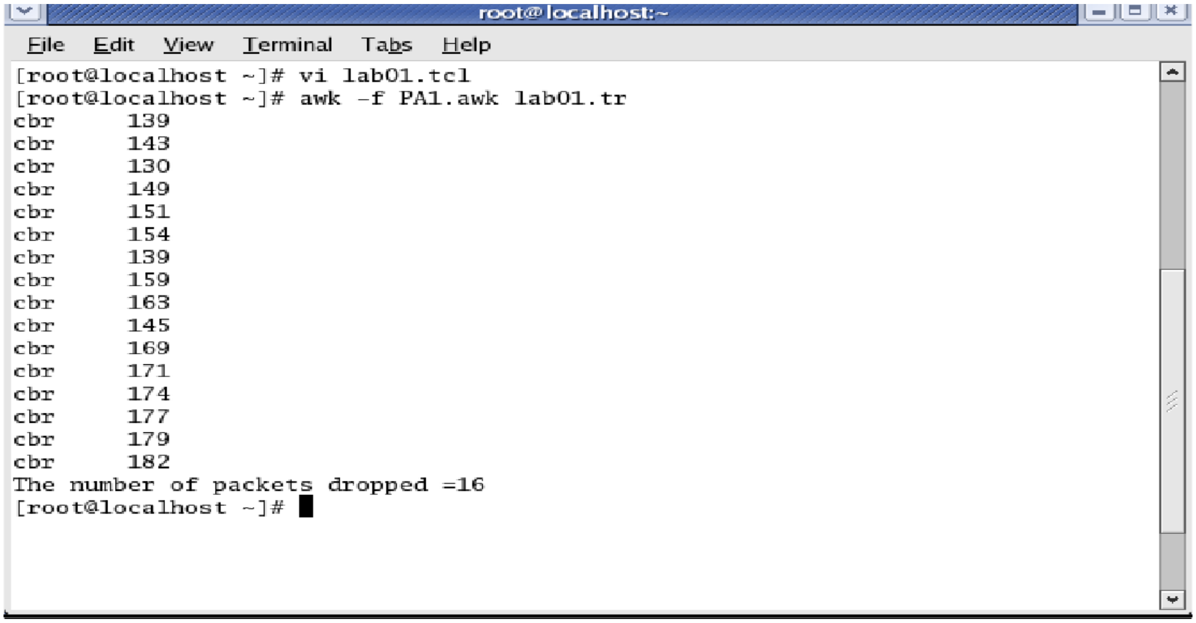
**Source Code :**

**pa1.awk**

```
BEGIN {
c=0;
}
{
if ($1=="d")
{
c++;
printf("%s\t%s\n",$5,$11);
}
}
END{
printf("The number of packet dropped= %d\n",c);
}
```



Auto layout: Ca 0.15 Cr 0.75 Iterations 10 Recalc



```
[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 ~]#
```

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

**Source Code:**

**pa2.tcl**

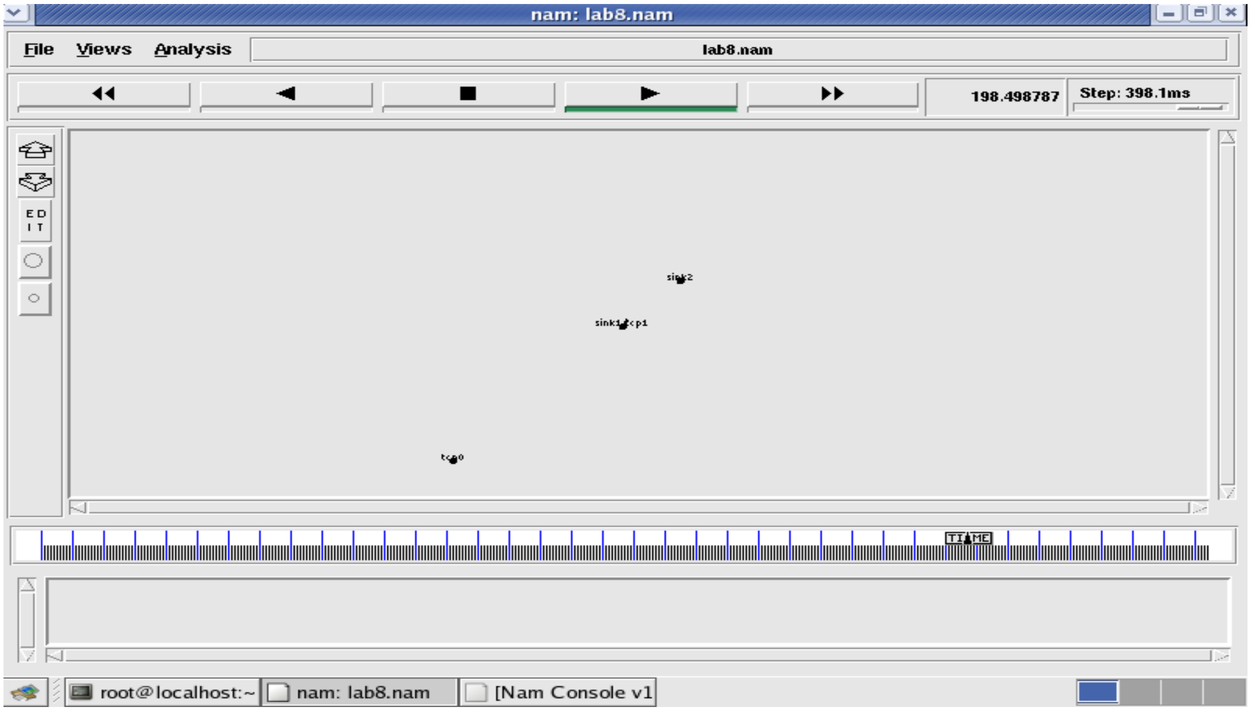
```
set ns [new Simulator]
set tf [open pa4.tr w]
$ns trace-all $tf
set topo [new Topography]
$topo load_flatgrid 1000 1000
set nf [open pa4.nam w]
$ns namtrace-all-wireless $nf 1000 1000
$ns node-config -adhocRouting DSDV \
                -llType LL \
                -macType Mac/802_11 \
                -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
```

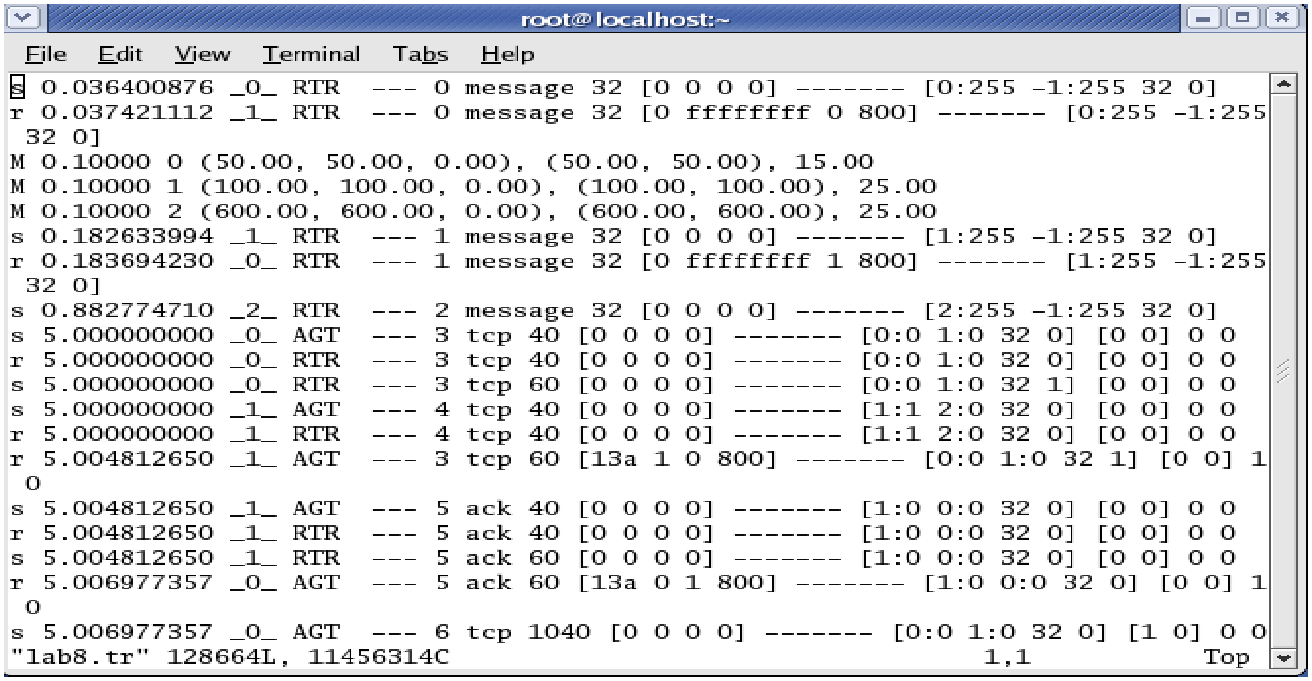
```
$ns at 0.1 "$n0 setdest 50 50 15"
$ns at 0.1 "$n1 setdest 100 100 25"
$ns at 0.1 "$n2 setdest 600 600 25"

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 pa4.nam &
close $tf
exit 0
}
$ns at 250 "finish"
$ns run
```

**<u>Source Code:</u>**

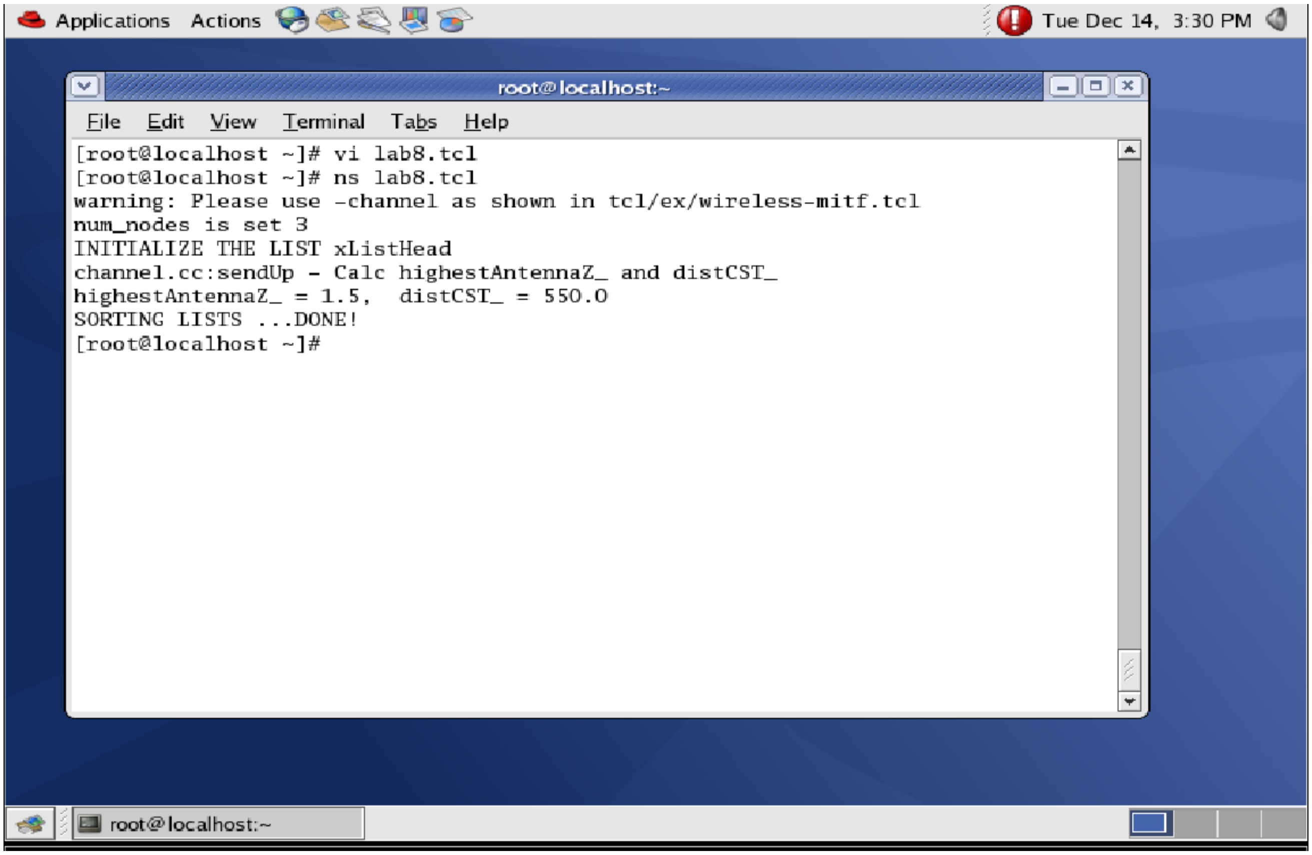
**pa2.awk**

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

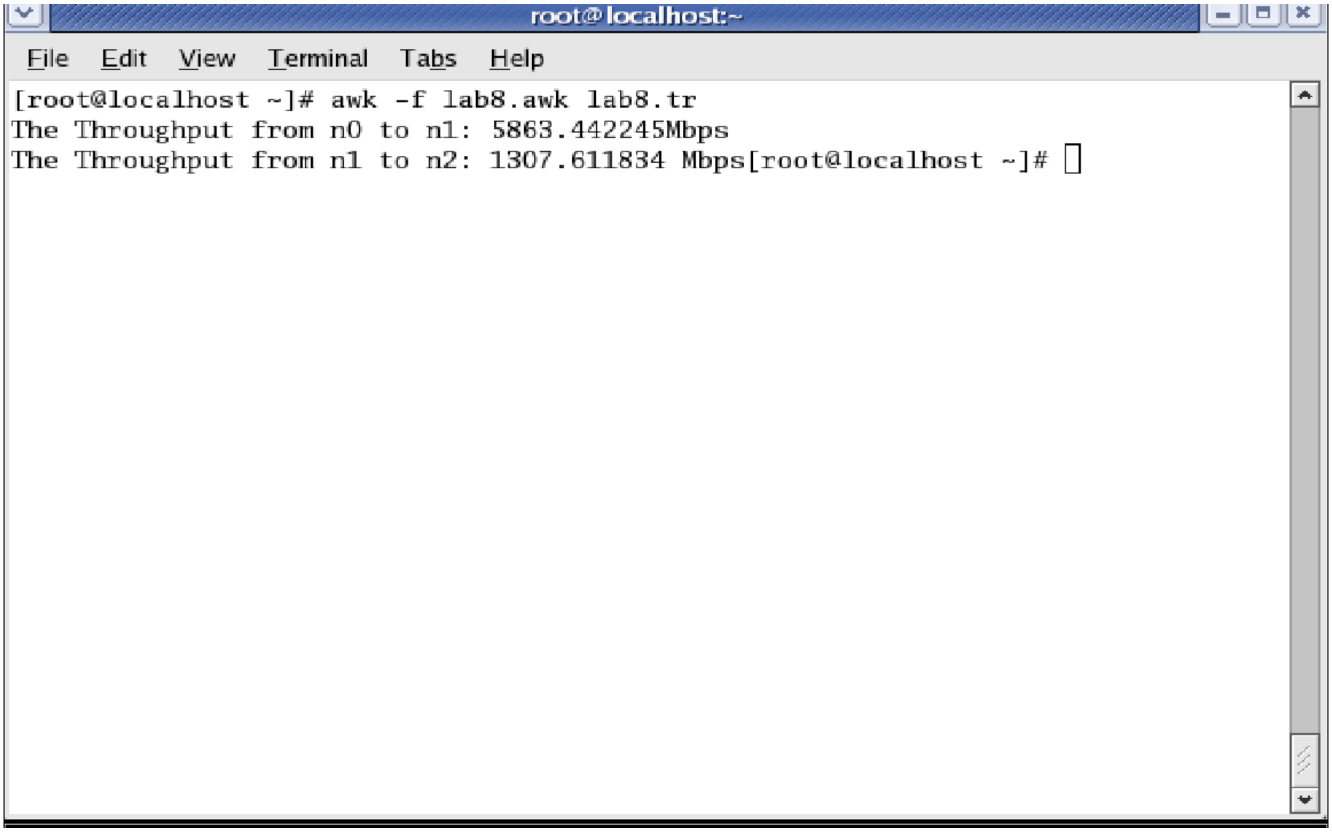




```
s 0.036400876 _0_ RTR  --- 0 message 32 [0 0 0 0] ------- [0:255 -1:255 32 0]
r 0.037421112 _1_ RTR  --- 0 message 32 [0 ffffffff 0 800] ------- [0:255 -1:255
 32 0]
M 0.10000 0 (50.00, 50.00, 0.00), (50.00, 50.00), 15.00
M 0.10000 1 (100.00, 100.00, 0.00), (100.00, 100.00), 25.00
M 0.10000 2 (600.00, 600.00, 0.00), (600.00, 600.00), 25.00
s 0.182633994 _1_ RTR  --- 1 message 32 [0 0 0 0] ------- [1:255 -1:255 32 0]
r 0.183694230 _0_ RTR  --- 1 message 32 [0 ffffffff 1 800] ------- [1:255 -1:255
 32 0]
s 0.882774710 _2_ RTR  --- 2 message 32 [0 0 0 0] ------- [2:255 -1:255 32 0]
s 5.000000000 _0_ AGT  --- 3 tcp 40 [0 0 0 0] ------- [0:0 1:0 32 0] [0 0] 0 0
r 5.000000000 _0_ RTR  --- 3 tcp 40 [0 0 0 0] ------- [0:0 1:0 32 0] [0 0] 0 0
s 5.000000000 _0_ RTR  --- 3 tcp 60 [0 0 0 0] ------- [0:0 1:0 32 1] [0 0] 0 0
s 5.000000000 _1_ AGT  --- 4 tcp 40 [0 0 0 0] ------- [1:1 2:0 32 0] [0 0] 0 0
r 5.000000000 _1_ RTR  --- 4 tcp 40 [0 0 0 0] ------- [1:1 2:0 32 0] [0 0] 0 0
r 5.004812650 _1_ AGT  --- 3 tcp 60 [13a 1 0 800] ------- [0:0 1:0 32 1] [0 0] 1
 0
s 5.004812650 _1_ AGT  --- 5 ack 40 [0 0 0 0] ------- [1:0 0:0 32 0] [0 0] 0 0
r 5.004812650 _1_ RTR  --- 5 ack 40 [0 0 0 0] ------- [1:0 0:0 32 0] [0 0] 0 0
s 5.004812650 _1_ RTR  --- 5 ack 60 [0 0 0 0] ------- [1:0 0:0 32 0] [0 0] 0 0
r 5.006977357 _0_ AGT  --- 5 ack 60 [13a 0 1 800] ------- [1:0 0:0 32 0] [0 0] 1
 0
s 5.006977357 _0_ AGT  --- 6 tcp 1040 [0 0 0 0] ------- [0:0 1:0 32 0] [1 0] 0 0
"lab8.tr" 128664L, 11456314C                                  1,1           Top
```



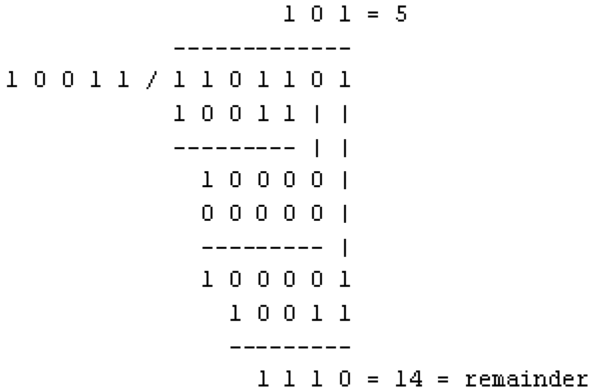
```
[root@localhost ~]# vi lab8.tcl
[root@localhost ~]# ns lab8.tcl
warning: Please use -channel as shown in tcl/ex/wireless-mitf.tcl
num_nodes is set 3
INITIALIZE THE LIST xListHead
channel.cc:sendUp - Calc highestAntennaZ_ and distCST_
highestAntennaZ_ = 1.5,  distCST_ = 550.0
SORTING LISTS ...DONE!
[root@localhost ~]#
```



```
[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 ~]#
```

### 3. Write a program for error detecting code using CRC-CCITT (16- bits).

Cyclic Redundancy Check Code for Error – Detection
The cyclic Redundancy Check (CRC) id a technique for detecting errors in data transmission, but not for correcting errors when they are detected.
CCITT- Consultative Committee for International Telegraphy and Telephone.



```
                    1 0 1 = 5
            -------------
1 0 0 1 1 / 1 1 0 1 1 0 1
            1 0 0 1 1 | |
            --------- | |
              1 0 0 0 0 |
              0 0 0 0 0 |
              --------- |
              1 0 0 0 0 1
                1 0 0 1 1
                ---------
                  1 1 1 0 = 14 = remainder
```

**Algorithm:**

**A) For Computing CRC:**

- The CRC Algorithm is based on polynomial arithmetic.
- Let the message that we have to send has k bits (denoted by M(x) in polynomial form having degree (k-1)) the sender and the receiver are agreed upon a generator polynomial having r bits (denoted by G(x) in polynomial form having degree (r-1)). The generator polynomial is also called “Divisor”.
- Now, append (r-1) zero bits to the LSB side of the message M(x) so it will now contain (k+r-1) bits and corresponds to the polynomial $x^{(r-1)}$ M (x).
- Divide the polynomial $x^{(r-1)}$ M (x) by Divisor, using modulo-2 subtraction (bit by bit XOR operation). Add the remainder R(x) (called frame check sequence) to $x^{(r-1)}$ M (x), using modulo -2 additions (bit by bit XOR operation). This is the message that will be transmitted by the transmitter denoted by T(x).

**B) For Error Detection:**

- Suppose that a transmission error occurs, so that the received message at the receiver is T(x) +E(x), instead of T(x). Each 1 bit in E(x) corresponds to a bit that has been inverted.
- The received message at the receiver end is divided by G(x), i.e. [T(x) + E(x) / G(x) ]. Since T(x)/G(x) is 0, so the result is simply E(x)/G(x).
- If E(x)/G(x) =0 than there is no error in the received message, otherwise there is an error.
- The following type of errors can be detected using CRC:
  - ➢ If G(x) has more than one bit and the coefficient of $x^0$ is 1, then all single bit errors are detected.

- If G(x) is not divisible by x ( the coefficient of $x^0$ is 1) , and t is the least positive integer (0<t<n-1) such that G(x) divides $x^1$ + 1, then all isolated double errors are detected.
- If G(x) has a factor (x+1), then all odd numbered errors are detected.

**Source code:**

```
import java.io.*;
class crc
{
public static void main(String[] args)
{
          InputStreamReader isr=new InputStreamReader (System.in);
          BufferedReader br=new BufferedReader(isr);
          int[] message;
          int[] gen;
          int[] app_message;
          int[] rem;
          int[] trans_message;
          int message_bits,gen_bits,total_bits;
          System.out.println("\n Enter the no of bits in the message");
          message_bits=Integer.parseInt(br.readLine());
          message=new int[message_bits];
          System.out.println("\n Enter the  message bits");
          for(int i=o;i<message_bits;i++)
          message[i]=Integer.parseInt(br.readLine());
          System.out.println("\n Enter the no of bits generated ");
          gen_bits=Integer.parseInt(br.readLine());
          gen=new int[gen_bits];
          System.out.println("\n Enter the generated  bits");
          for(int i=o;i<gen_bits;i++);
          {
          gen[i]=Integer.parseInt(br.readLine());
          }
          total_bits=message_bits+gen_bits-1;
          app_message=new int[total_bits];
          rem=new int[total_bits];
          trans_message=new int[total_bits];
          for(int i=0;i<message.length;i++)
          {
          app_message[i]=message[i];
          }
          System.out.print("\n Message bits :");
          for(int i=o;i<message_bits;i++)
          {
          System.out.print(message[i]);
          }
          System.out.print("\n Generated  bits :");
          for(int i=0;i<gen_bits;i++)
          {
          System.out.print(gen[i]);
```

```
}
System.out.print("\n Appended message :");
for(int i=0;i<app_message.length;i++)
{
System.out.print(app_message[i]);
}
for(int j=0;j<app_message.length;j++)
{
        rem[j]=app_message[j];
}
rem=computecrc(app_message.length;i++)
for(int i=0;i<app_message.length;i++)
{
trans_mesage[i]=(app_message[i]^rem[i]);
}
System.out.println("\n Transmitted message from the transmitter  :");
for(int i=0;i<trans_message.length;i++)
{
System.out.print(trans_message[i]);
}
System.out.println("\n Message of"+total_bits+"bits received");
for(int i=0;i<trans_message.length;i++)
{
trans_messge[i]=Inter.parseInt(br.readLine());
}
System.out.println("\n Recevied message is :");
for(int i=0;i<trans_message.length;i++)
{
System.out.print(trans_message[i]);
}
for(int j=0;j<trans_message.length;j++)
{
rem[j]=trans_message[j];
}
rem=computercrc(trans_message,gen,rem);
for (int i=0;i<rem.length;i++)
{
if(rem[i]!=0)
{
System.out.println("\n There is error");
break;
}
if(i==rem.length-1)
{
System.out.println("\n There is no error");
```

```
            }
}

static int[] computecrc(int app_message[],int gen[],int rem[])
{
            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;
}
}
}
```

**Out Put:**

```
Enter the no of bits in the message : 4
Enter the message bits : 1 0 0 1
Enter the no of bits generated : 17
Enter the generated bits :
1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1
Message bits: 1 0 0 1
Generated bits : 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1
Appended message : 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Transmitted message from transmitter : 1 0 0 1  1 0 0 1 0 0 0 1 0 0 1 0 1 0 0 1
Message of 20 bits received :
1 0 0  1 1 0 0 1 0 0 0 1 0 0 1 0 1 0 0 1
There is no error


Enter the no of bits in the message : 4
Enter the message bits : 1 0 0 1
Enter the no of bits generated : 17
Enter the generated bits :
1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1
Message bits: 1 0 0 1
Generated bits : 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1
Appended message : 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Transmitted message from transmitter : 1 0 0 1  1 0 0 1 0 0 0 1 0 0 0 1 0 1 0 0 1
Message of 20 bits received :
1 0 0 1 1 0 0 1 0 1 0 1 0 0 1 0 1 0 1 0
There is an error
```

**4. 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 in the network.**

**Source Code:**

**pa4.tcl**

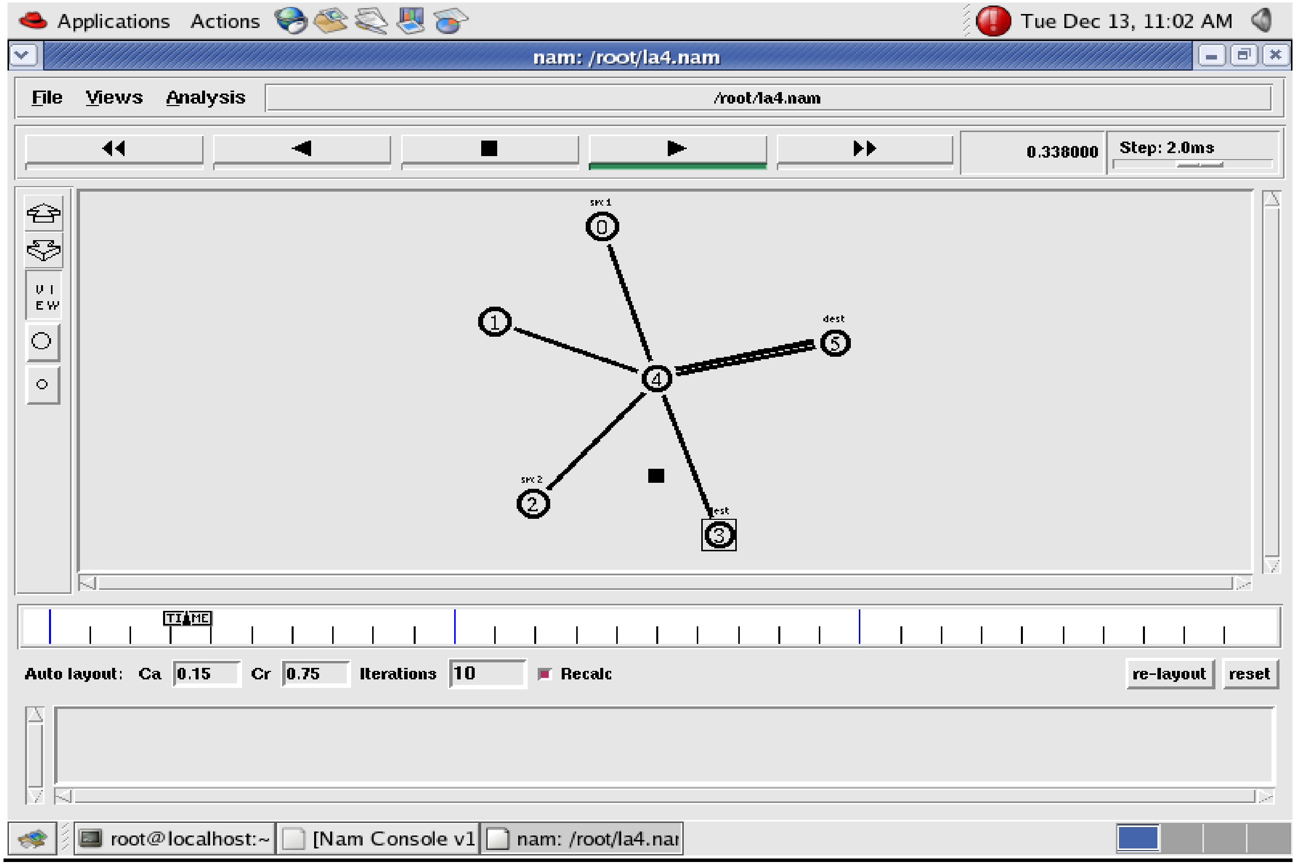
```
set ns [ new Simulator ]
set nf [ open pa2.nam w ]
$ns namtrace-all $nf
set tf [ open pa2.tr w ]
$ns trace-all $tf
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
$n4 shape box
$ns duplex-link $n0 $n4 1005Mb 1ms DropTail
$ns duplex-link $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_ 500000
$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"
}
$ns connect $p1 $p5
$ns connect $p3 $p4
proc finish { } {
global ns nf tf
$ns flush-trace
close $nf
close $tf
exec nam pa2.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
```

**Source Code :**

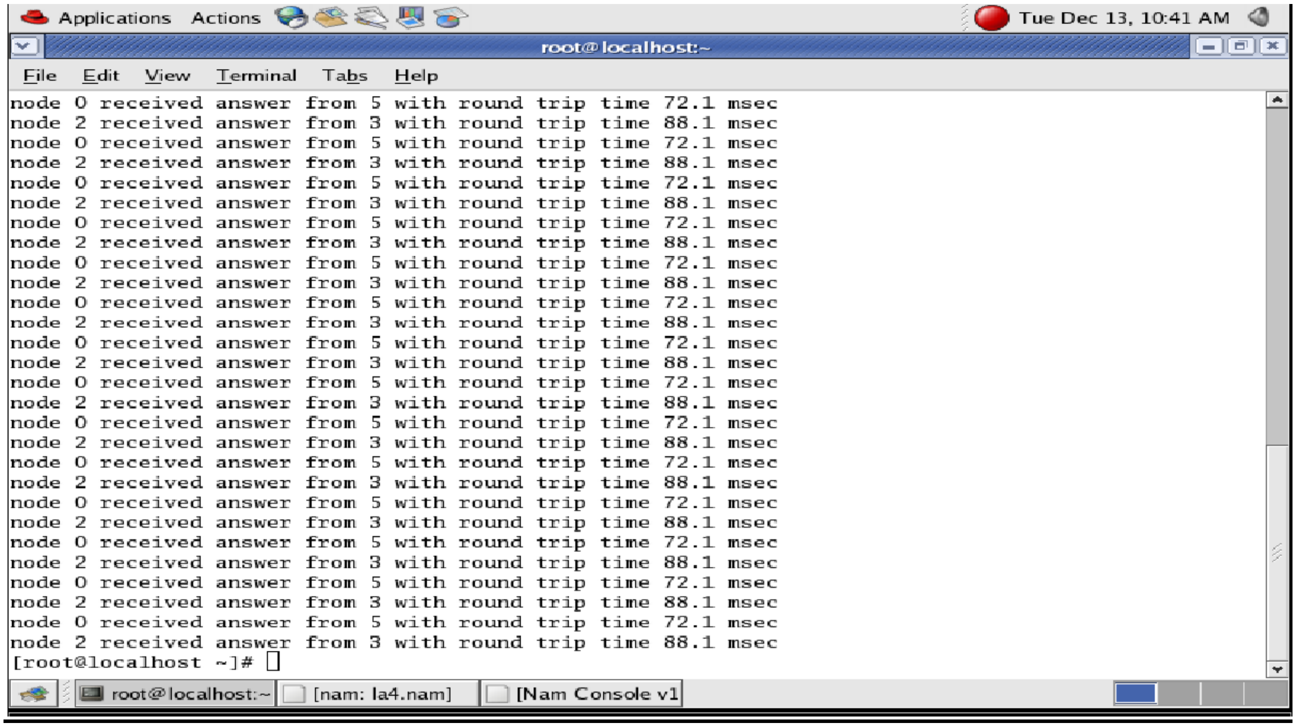
**pa4.awk**

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

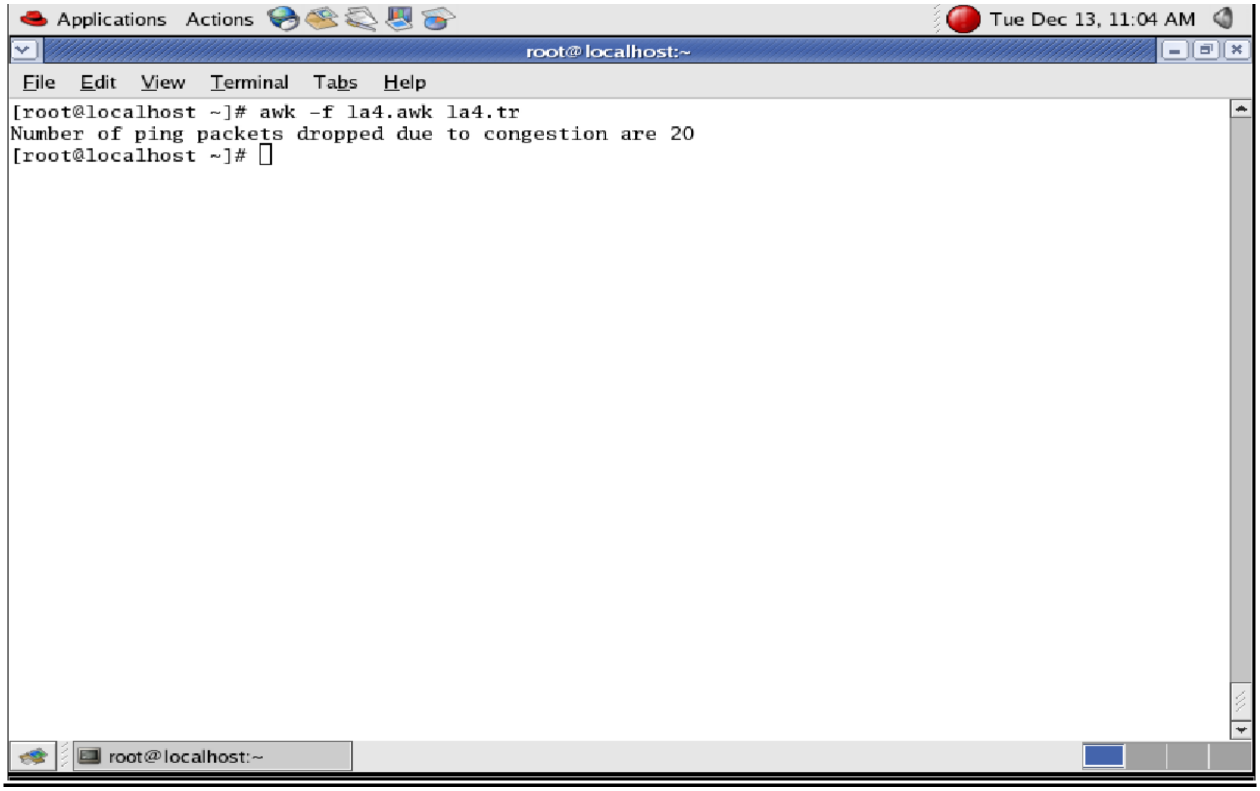


0.338000 Step: 2.0ms

Auto layout: Ca 0.15 Cr 0.75 Iterations 10 ☒ Recalc



```
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 ~]#
```

## 5. Write a program to find the shortest path between vertices using bellman-ford algorithm.

Bell man ford algorithm is a procedure used to find all shortest path in a graph from one source to all other nodes. The algorithm was introduced by American mathematicians Richard Bellman and Lester ford. Computes shortest from a single source vertex to all of the other vertices in a weighted diagraph.

**<u>Source Code</u>:**

```
import java.io.*;
import java.util.*;
import java.io.DataInputStream;
class Edge
{
    int source;
    int dest;
    int weight;
}
class Bellman
{
    public static void BellmanFord(Edge edges[], int edgecount, int nodecount, int source)
    {
            int infinity=50000;
            int i, j;
            int distance[]=new int[nodecount];
            for(i=0; i&lt;nodecount; i++)
            distance[i]=infinity;
            distance[source]=0;
            for(i=0; i&lt;nodecount; i++)
            {
                    boolean somethingchanged=false;
                    for(j=0; j&lt;edgecount; j++)
                    {
                            if(distance[edges[j].source]!=infinity)
                            {
                                int new_distance=distance[edges[j].source]+edges[j].weight;
                                if(new_distance&lt;distance[edges[j].dest])
                                {
                                    distance[edges[j].dest]=new_distance;
                                    somethingchanged=true;
```

```
                  }
              }
          }
          if(!somethingchanged)
          break;
          }
          for(i=0; i&lt;edgecount; ++i)
          {
          if(distance[edges[i].dest]&gt;distance[edges[i].source]+edges[i].weight)
        System.out.println(&quot;Negative edge weight cycles detected!!!&quot;);
          }
for(i=0; i&lt;nodecount; ++i)
System.out.println(&quot;The shortest distance between nodes &quot;+source+&quot; &amp;
&quot;+i+&quot; is &quot;+distance[i]);
}
public static void main(String args[])
{
Scanner in=new Scanner(System.in);
Edge edges[]=new Edge[10];
for( int i=0; i&lt;10; i++)
{
edges[i]=new Edge();
System.out.print(&quot;Enter source number [&quot;+i+&quot;] : &quot;);
edges[i].source=in.nextInt();
System.out.print(&quot;Enter destination number [&quot;+i+&quot;] : &quot;);
edges[i].dest=in.nextInt();
System.out.print(&quot;Enter weight number [&quot;+i+&quot;] : &quot;);
edges[i].weight=in.nextInt();
System.out.println();
}
BellmanFord(edges, 10, 5, 4);
}
}
```

**Out Put:**

```
Enter the number of vertices : 4
Enter the adjacency matrix:
0      5      999     999
5      0      2      4
999    2      0      1
999    4      1      0
Enter source of vertex: 1
Distance of source 1 to 1 is 0
Distance of source 1 to 2 is 5
Distance of source 1 to 3 is 7
Distance of source 1 to 4 is 8



Enter the number of vertices : 6
Enter the adjacency matrix:
0      10     3      999    999    999
999    0      999    -3     2      999
999    999    0      3      999    999
999    999    999    0      4      1
999    -1     999    999    0      999
999    999    999    999    4      0
Enter source of vertex: 1
Distance of source 1 to 1 is 0
 Distance of source 1 to 2 is 9
Distance of source 1 to 3 is 3
Distance of source 1 to 4 is 6
Distance of source 1 to 5 is 10
Distance of source 1 to 6 is 7
```

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

**Source Code:**

**pa6.tcl**

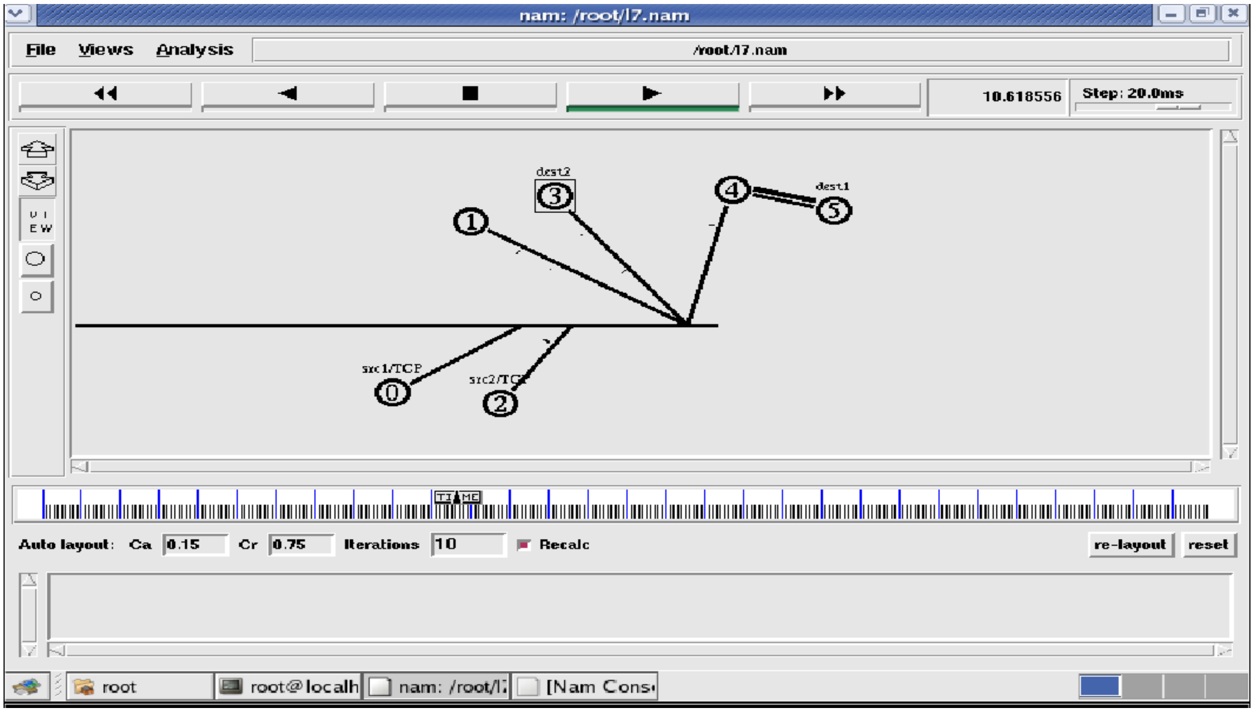
```
set ns [new Simulator]
set tf [open pa3.tr w]
$ns trace-all $tf
set nf [open pa3.nam w]
$ns namtrace-all $nf

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]
```

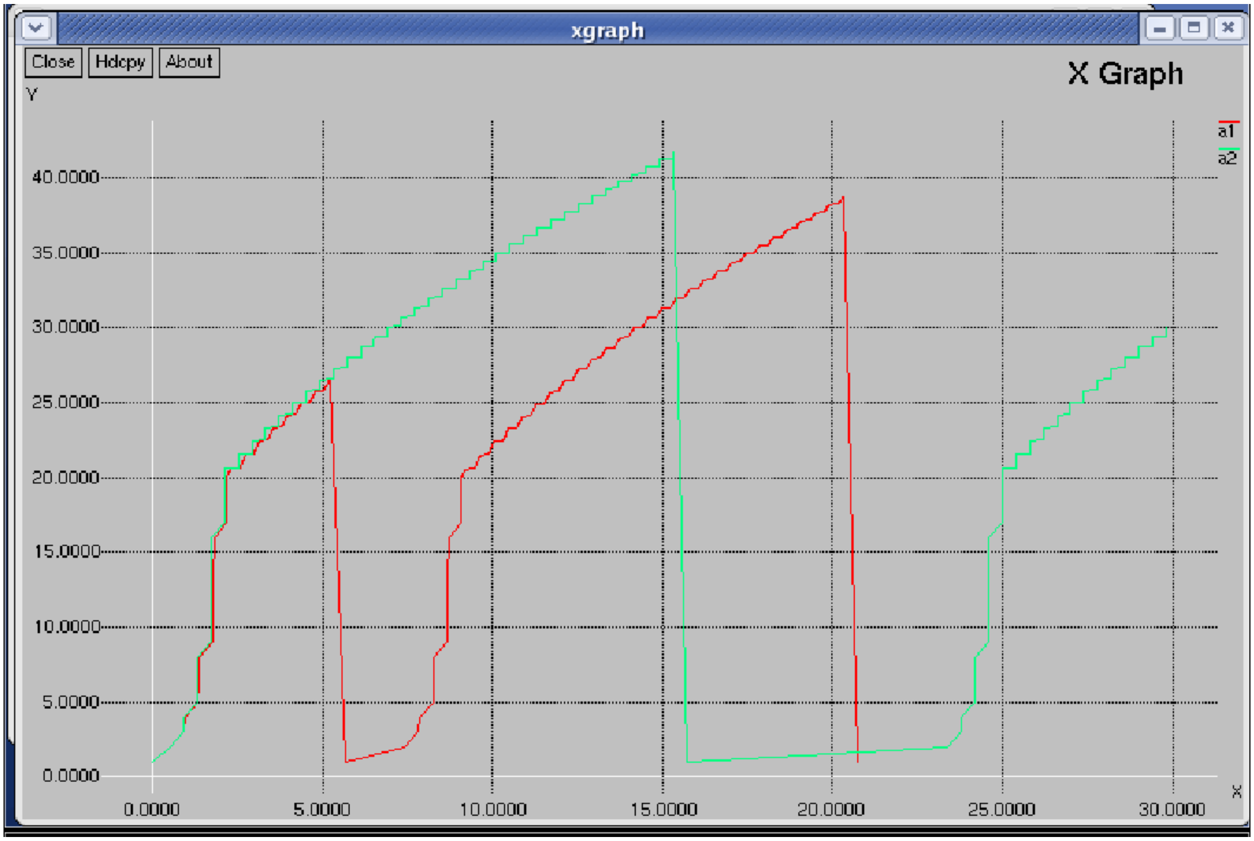
```
$ftp2 attach-agent $tcp2
$ftp2 set packetSize_ 600
$ftp2 set interval_ 0.0001
set sink3 [new Agent/TCPSink]
$ns attach-agent $n3 $sink3

$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 pa3.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
```

**Source Code:**

**pa6.awk**

```
BEGIN {
}
{
if($6=="cwnd_")
printf("%f\t%f\t\n",$1,$7);
}
END{
}
```


nam: /root/l7.nam
File
Views
Analysis
/root/l7.nam
10.618556
Step: 20.0ms
dest2
dest1
src1/TCP
src2/TCP
0
1
2
3
4
5
Auto layout:
Ca 0.15
Cr 0.75
Iterations 10
Recalc
re-layout
reset
root
root@localh
nam: /root/l7
[Nam Cons


xgraph
Close
Hdcpy
About
X Graph
Y
X
a1
a2
40.0000
35.0000
30.0000
25.0000
20.0000
15.0000
10.0000
5.0000
0.0000
0.0000
5.0000
10.0000
15.0000
20.0000
25.0000
30.0000

## 7. Write a program for congestion control using leaky bucket algorithm.

What is **congestion**?
A state occurring in network layer when the message traffic is so heavy that it slows down network response time.
**Effects** of Congestion

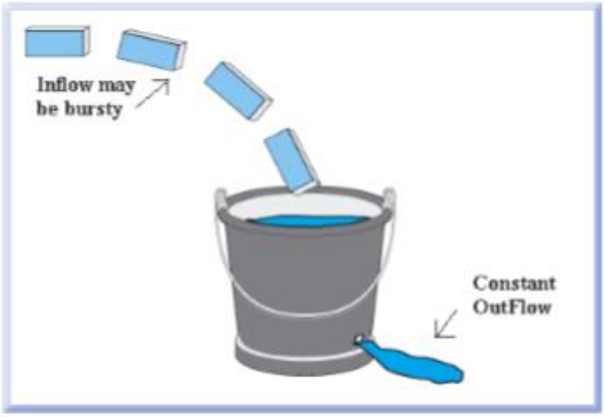
- As delay increases, performance decreases.
- If delay increases, retransmission occurs, making situation worse.

Traffic shaping or policing: To control the amount and rate of traffic is called Traffic shaping or policing. Traffic shaping term is used when the traffic leaves a network. Policing term is used when the data enters the network. Two techniques can shape or police the traffic leaky bucket and token bucket.

**Congestion control algorithms**

- **Leaky Bucket Algorithm**

Let us consider an example to understand
Imagine a bucket with a small hole in the bottom. No matter at what rate water enters the bucket, the outflow is at constant rate. When the bucket is full with water additional water entering spills over the sides and is lost.



Similarly, each network interface contains a leaky bucket and the following **steps** are involved in leaky bucket algorithm:

1. When host wants to send packet, packet is thrown into the bucket.
2. The bucket leaks at a constant rate, meaning the network interface transmits packets at a constant rate.
3. Bursty traffic is converted to a uniform traffic by the leaky bucket.
4. In practice the bucket is a finite queue that outputs at a finite rate.

**Source Code:**

```
importjava.util.*;
public class leaky
{
    public static void main(String[] args)
{
    Scanner my = new Scanner(System.in);
    intno_groups,bucket_size;
    System.out.print("\n Enter the bucket size : \t");
    bucket_size = my.nextInt();
    System.out.print("\n Enter the no of groups : \t");
    no_groups = my.nextInt();
    intno_packets[] = new int[no_groups];
    intin_bw[] = new int[no_groups];
    intout_bw,reqd_bw=0,tot_packets=0;
    for(inti=0;i<no_groups;i++)
    {
        System.out.print("\n Enter the no of packets for group " + (i+1) + "\t");
        no_packets[i] = my.nextInt();
        System.out.print("\n Enter the input bandwidth for the group " + (i+1) + "\t");
        in_bw[i] = my.nextInt();
        if((tot_packets+no_packets[i])<=bucket_size)
        {
            tot_packets += no_packets[i];
        }
        else
         {
            do
        {
            System.out.println(" Bucket Overflow ");
            System.out.println(" Enter value less than " + (bucket_size-tot_packets));
            no_packets[i] = my.nextInt();
            }while((tot_packets+no_packets[i])>bucket_size);
            tot_packets += no_packets[i];
         }
        reqd_bw += (no_packets[i]*in_bw[i]);
    }
    System.out.println("\nThe total required bandwidth is " + reqd_bw);
    System.out.println("Enter the output bandwidth ");
    out_bw = my.nextInt();
    int temp=reqd_bw;
    intrem_pkts = tot_packets;
    while((out_bw<=temp)&&(rem_pkts>0))
    {
        System.out.println("Data Sent \n" + (--rem_pkts) + " packets remaining");
```

```
                        System.out.println("Remaining Bandwidth " + (temp -= out_bw));
                        if((out_bw>temp)&&(rem_pkts>0))
                        System.out.println(rem_pkts + " packet(s) discarded due to insufficient bandwidth");
            }
    }
}
```

**Out Put:**

```
Enter bucket size : 20
Enter number of groups :2
Enter number of packets of group 1:2
Enter input bandwidth for group 1:2
Enter number of packets of group 2 :3
Enter input bandwidth for group 2 : 3
Total required bandwidth :13
Enter the output bandwidth:2
Data sent
4 packets remaining
Remaining bandwidth:11
Data sent
3 packets remaining
Remaining bandwidth : 9
Data sent
2 packets remaining
Remaining bandwidth :7
Data sent
1 packets remaining
Remaining bandwidth :5
Data sent
0 packets remaining
Remaining bandwidth :3
```

**Viva Questions:**

1. **What are functions of different layers?**
2. **Differentiate between TCP/IP Layers and OSI Layers**
3. **Why header is required?**
4. **What is the use of adding header and trailer to frames?**
5. **What is encapsulation?**
6. **Why fragmentation requires?**
7. **What is MTU?**
8. **Which layer imposes MTU?**
9. **Differentiate between flow control and congestion control.**
10. **Differentiate between Point-to-Point Connection and End-to-End connections.**
11. **What are protocols running in different layers?**
12. **What is Protocol Stack?**
13. **Differentiate between TCP and UDP.**
14. **Differentiate between Connectionless and connection oriented connection.**
15. **Why frame sorting is required?**
16. **What is meant by subnet?**
17. **What is meant by Gateway?**
18. **What is an IP address?**
19. **What is MAC address?**
20. **Why IP address is required when we have MAC address?**
21. **What is meant by port?**
22. **What are ephemerical port number and well known port numbers?**
23. **What is a socket?**
24. **What are the parameters of socket()?**
25. **Describe bind(), listen(), accept(), connect(), send() and recv().**
26. **What are system calls? Mention few of them.**
27. **What is meant by file descriptor?**
28. **What is meant by traffic shaping?**
29. **How do you classify congestion control algorithms?**
30. **How do you implement Leaky bucket?**
31. **How do you generate busty traffic?**
32. **What is the polynomial used in CRC-CCITT?**
33. **. What are the other error detection algorithms?**
34. **What are Routing algorithms?**
35. **How do you classify routing algorithms? Give examples for each.**
36. **What are drawbacks in distance vector algorithm?**
37. **How routers update distances to each of its neighbor?**
38. **How do you overcome count to infinity problem?**
39. **What is cryptography?**

40. **How do you classify cryptographic algorithms?**
41. **What is public key?**
42. **What is private key?**
43. **What are key cipher text and plaintext?**
44. **What is simulation?**
45. **What are advantages of simulation?**
46. **Differentiate between Simulation and Emulation.**
47. **What is meant by router?**
48. **What is meant by bridge?**
49. **What is meant by switch?**
50. **What is meant by hub?**
51. **Differentiate between route, bridge, switch and hub.**
52. **What is ping and telnet?**
53. **What is FTP?**
54. **What is collision?**
55. **How do you generate multiple traffics across different sender-receiver pairs?**
56. **What is meant by mobile host?**
57. **Name few other Network simulators**
58. **Differentiate between logical and physical address.**
59. **Which address gets affected if a system moves from one place to anotherplace?**
60. **What is ICMP? What are uses of ICMP? Name few.**
61. **Which layer implements security for data?**
62. **What is connectionless and connection oriented protocol?**
63. **What is Congestion window?**
64. **What is flow control?**