10 YEARS OF UNIVERSITY RECOGNITION

20 YEARS OF ACADEMIC EXCELLENCE

REVA UNIVERSITY
Bengaluru, India

School of Electronics and Communication Engineering

Program
B.Tech. in ECE

Computer Networks Lab

LABORATORY MANUAL

B20EN0606

VI Semester
2020-24

Rukmini Educational Charitable Trust

www.reva.edu.in

Vision of the University

''REVA University aspires to become an innovative university by developing excellent human resources with leadership qualities, ethical and moral values, research culture and innovative skills through higher education of global standards"

Mission of the University

- To create excellent infrastructure facilities and state-of-the-art laboratories and incubation centres
- To provide student-centric learning environment through innovative pedagogy and education reforms
- To encourage research and entrepreneurship through collaborations and extension activities
- To promote industry-institute partnerships and share knowledge for innovation and development
- To organize society development programs for knowledge enhancement in thrust areas
- To enhance leadership qualities among the youth and enrich personality traits, promote patriotism and moral values.

Vision of the School

The School of Electronics and Communication Engineering is envisioned to be a leading centre of higher learning with academic excellence in the field of electronics and communication engineering blended by research and innovation in tune with changing technological and cultural challenges supported with leadership qualities, ethical and moral values.

Mission of the School

- Establish a unique learning environment to enable the students to face the challenges in the field of Electronics and Communication Engineering and explore multidisciplinary which serve the societal requirements.
- Create state-of-the-art laboratories, resources, and exposure to the current industrial trends to enable students to develop skills for solving complex technological problems of current times and provide a framework for promoting collaborative and multidisciplinary activities.
- Promote the establishment of Centres of Excellence in niche technology areas to nurture the spirit of innovation and creativity among faculty and students.
- Offer ethical and moral value-based education by promoting activities which inculcate the leadership qualities, patriotism and set high benchmarks to serve the society

Program Educational Objectives (PEOs)

The Program Educational Objectives of B. Tech in B.Tech in ECE/ ECM Engineering are as follows:

PEO -1: To have successful professional careers in industry, government, academia, and military as innovative engineers.

PEO -2: To successfully solve engineering problems associated with the lifecycle of B.Tech in ECE/ ECM Engineering Systems either leading a team or as a team member.

PEO -3: To continue to learn and advance their careers through activities such as participation in professional organizations, attainment of professional certification for lifelong learning and seeking higher education.

PEO -4: To be active members ready to serve the society locally and internationally and will undertake entrepreneurship for the growth of economy and to generate employment.

Program Outcomes (POs)

On successful completion of the program, the graduates of B. Tech. in ECE/ ECM Engineering program will be able to:

- **PO-1: Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals for the solution of complex problems in B.Tech in ECE/ ECM Engineering.
- **PO-2: Problem analysis:** Identify, formulate, research literature, and analyze engineering problems to arrive at substantiated conclusions using first principles of mathematics, natural, and engineering sciences.
- **PO-3: Design/development of solutions:** Design solutions for complex engineering problems and design system components, processes to meet the specifications with consideration for the public health and safety, and the cultural, societal, and environmental considerations.
- **PO-4: Conduct investigations of complex problems:** Use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
- **PO-5: 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.
- **PO-6: 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.
- **PO-7: 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.
- **PO-8: Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
- **PO-9: Individual and team work:** Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.
- **PO-10: Communication:** Communicate effectively with the engineering community and with society at large. Be able to comprehend and write effective reports documentation. Make effective presentations, and give and receive clear instructions.
- **PO-11: Project management and finance:** Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work, as a member and leader in a team. Manage projects in multidisciplinary environments.
- **PO-12: 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.

CONTENTS

Course Title	Computer Networks Lab				Course Type		HC	
Course Code	B20EN0606	Credits	1		Class		VI Semester	
Course Description	LTP	Credits	Contact Hours	Work Load	Total Number of Classes Per Semester		Assessment in Weightage	
	Lecture	-	-	-				
	Tutorial	-						
	Practice	1	2	2	Theory	Practical	IA	SEE
	-	-	-	-				
	Total	1	2	2		28	50 %	50 %

COURSE OBJECTIVES:

The objectives of this course are to:

1. Identify the necessary software and hardware to constitute a designed computer network
2. Implement a simple LAN Network
3. Describe, Analyze and evaluate a number of datalink, network, and transport layer protocols
4. Describe routing protocols

COURSE OUTCOMES (COs)

On successful completion of this course; the student shall be able to:

CO#	Course Outcomes	POs	PSOs
CO1	Write and debug the code for various error detection, Congestion Control Techniques		
CO2	Write and test the code using different security techniques to secure the messages,		
CO3	Write the program and Evaluate different network layer and transport layer protocols		
CO4	Write the code for different wired and wireless network scenarios and test the performance using simulators		
CO5	Evaluate various design parameters such as latency, error rate, throughput, and their influence on node/link utilization and performance		

BLOOM'S LEVEL OF THE COURSE OUTCOMES

CO#	Bloom's Level					
	Remember (L1)	Understand (L2)	Apply (L3)	Analyze (L4)	Evaluate (L5)	Create (L6)
CO1	✓	✓	✓	✓	✓	
CO2	✓	✓	✓	✓	✓	
CO3	✓	✓	✓	✓	✓	
CO4	✓	✓	✓	✓	✓	
CO5	✓	✓	✓	✓	✓	
CO6	✓	✓	✓	✓	✓	

COURSE ARTICULATION MATRIX

CO#/ POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	2	2	2		3				2	2			1	2	3
CO2	2	2	2		3				2	2			1	2	3
CO3	2	2	2		3				2	2			1	2	3
CO4	2	2	2		3				2	2			1	2	3
CO5	2	2	2		3				2	2			1	2	3
CO6	2	2	2		3				2	2			1	2	3

Note:1-Low,2-Medium,3-High

PRACTICE SESSION:

Sl. No.	Name of the Practice Session	Tools and Techniques	Expected Skill /Ability
1	Write a program for bit stuffing & de-stuffing using HDLC.	C/C++ Software	Identify bit stuffing and destuffing
2	Write a program for character stuffing & de-stuffing using HDLC.	C/C++ Software	Identify byte stuffing and destuffing
3	Perform the Encryption and Decryption of a given message using substitution method.	C/C++ Software	Analyze the Encryption and Decryption of a given message using substitution method.
4	Choose the two prime numbers, p=17 and q=11.Write a program for public key encryption system using RSA algorithm to encrypt and decrypt the message.	C/C++ Software	Understand the key concept of public key encryption system using RSA algorithm to encrypt and decrypt the message.
5	Write a program to implement the congestion control b using the leaky bucket algorithm. Examine node transmitting/receiving packets to/from other nodes. Using a random function; vary the packet size.	C/C++ Software	Analyze the leaky bucket algorithm for congestion control
6	Write a program for distance vector algorithm to find the shortest path for transmission.	C/C++ Software	Analyze to find the shortest path using the distance vector algorithm
7	Create a three node network topology and connect the duplex links between them. Tcl script to observe the packet flow for the given network in network animator (NAM)	NS2 Simulator Software	Understand the concept of duplex link in a given three node topology, and analyze the packet flow.
8	Simulate a four node point-to-point network, and connect the links as follows: n0-n2, n1-n2 and n2-n3. Apply TCP agent between n0-n3, n1-n3. Apply relevant applications over TCP agents by changing the parameters and hence determine the number of packets transmitted.	NS2 Simulator Software	Analyze the concept of TCP agent for a given four node network and determine the number of packets transmitted
9	Simulate a four node point-to-point network, and connect the links as follows: n0-n2, n1-n2 and n2-n3. Apply UDP agent between n0-n3, n1-n3. Apply relevant applications over UDP agents by changing the parameters and hence determine the number of packets transmitted.	NS2 Simulator Software	Analyze the concept of UDP agent for a given four node network and determine the number of packets transmitted
10	Simulate a three nodes point-to point network and connect the duplex links between them. Set the queue size, vary the transmission speeds (bandwidth)and find the number of packets dropped.	NS2 Simulator Software	Evaluate the concept of duplex link in a given three node topology, and analyze queue size, the transmission speeds (bandwidth)and the number of packets dropped.

11	Simulate an Ethernet LAN using N-nodes (6-10) with UDP/TCP connection. Apply relevant applications over UDP/TCP agents by changing the parameters and hence determine the number of packets transmitted.	NS2 Simulator Software	Analyze packet transmission in 802.3 Ethernet using UDP/TCP
12	Simulate a wireless network for n nodes. For a wireless network consisting of three mobile nodes (n0-n2), Nodes are configured with the specific parameters of a wireless node. Initial location of the node is fixed. Nodes are given mobility with fixed speed and fixed destination location. TCP agent is attached to node0 and TCP sink agent is attached to node1. Both the agents are connected and FTP application is attached to TCP agent. Write a Tcl script and make an ad-hoc simulation to analyze the output in the trace file. Use the routing protocol as Adhoc on demand distance vector (AODV).	NS2 Simulator Software	Design and analyze AODV protocol for wireless networks.

TEXT BOOKS:

1. B Forouzan "Data Communication and Networking", 4th Ed, TMH 2006.

REFERENCE BOOKS:

1. James F. Kurose, Keith W. Ross "Computer Networks", Pearson Education, 2nd Edition, 2003.
2. Wayne Tomasi"Introduction to Data communication and Networking" Pearson Education 2007.

S. Keshav, "An Engineering Approach to Computer Networking", Pearson Education

Course Contents:

PART A – Simulation Exercises

1. Write a program for bit stuffing & de-stuffing using HDLC.
2. Write a program for character stuffing & de-stuffing using HDLC.
3. Perform the Encryption and Decryption of a given message using substitution method.
4. Choose the two prime numbers, p=17 and q=11.Write a program for public key encryption system using RSA algorithm to encrypt and decrypt the message.
5. Write a program to implement the congestion control-b using the leaky bucket algorithm. Examine node transmitting/receiving packets to/from other nodes. Using a random function; vary the packet size.
6. Write a program for distance vector algorithm to find the shortest path for transmission.

PART- B

1. Create a three node network topology and connect the duplex links between them. TCL script toobserve the packet flow for the given network in network animator (NAM).

2. Simulate a four node point-to-point network, and connect the links as follows: n0-n2, n1-n2 andn2-n3. Apply TCP agent between n0-n3, n1-n3. Apply relevant applications over TCP agents by changing the parameters and hence determine the number of packets transmitted.

3. Simulate a four node point-to-point network, and connect the links as follows: n0-n2, n1-n2 andn2-n3. Apply UDP agent between n0-n3, n1-n3. Apply relevant applications over UDP agents by changing the parameters and hence determine the number of packets transmitted.

4. Simulate a three nodes point-to-point network and connect the duplex links between them. Set the queue size, vary the transmission speeds (bandwidth) and find the number of packets dropped. (Point to point network with the links connected as follows: n0 – n2, n1 – n2 and n2 – n3. Apply TCP agent between n0 – n3 and UDP agent between n1 – n3. Apply relevant applications over TCP and UDP agents changing the parameter and determine the number of packets sent by TCP / UDP).

5. Simulate an Ethernet LAN using N-nodes (6-10) with UDP/TCP connection. Apply relevant applications over UDP/TCP agents by changing the parameters and hence determine the number of packets transmitted.

6. Simulate a wireless network for n nodes. For a wireless network consisting of three mobile nodes (n0-n2), Nodes are configured with the specific parameters of a wireless node. Initial location of the node is fixed. Nodes are given mobility with fixed speed and fixed destination location. TCP agent is attached to node0 and TCP sink agent is attached to node1. Both the agents are connected and FTP application is attached to TCP agent. Write a Tcl script and make an ad-hoc simulation to analyze the output in the trace file. Use the routing protocol as Adhoc on demanddistance vector (AODV).

Simulator: Any of the following simulators can be used.

1. NS2 Simulator (http://www.isi.edu/nsnam/ns/)
2. NS3 Simulator (https://www.nsnam.org/)
3. NCTUns (http://csie.nqu.edu.tw/smallko/nctuns/nctuns.htm)
4. BOSON simulator (http://www.boson.com/)

Part A – (Programs on different Layers)

1. Write a program for bit stuffing & de-stuffing using HDLC.
2. Write a program for character stuffing & de-stuffing using HDLC.
3. Perform the Encryption and Decryption of a given message using substitution method.
4. Choose the two prime numbers, p=17 and q=11.Write a program for public key encryption system using RSA algorithm to encrypt and decrypt the message.
5. Write a program to implement the congestion control-b using the leaky bucket algorithm. Examine node transmitting/receiving packets to/from other nodes. Using a random function; vary the packet size.
6. Write a program for distance vector algorithm to find the shortest path for transmission

PART A – C Programming

Experiment 1: Simulate bit stuffing and de-stuffing Using HDLC

Aim: To write and execute program for bit stuffing and de-stuffing

Theory: When a data stream is transmitted over a channel it may get deviated due to the noise, interference and other nonlinear Characteristics of the nature. Therefore for a purpose of error Detection and correction the ordinary raw data is divided into Frames and compute the checksum for that.

Checksum: Error detection method used by the higher level protocol check sum generator divides the data into equal segments of n bits, add the segments and complement the result called checksum field is added to the end of the original data unit as redundancy bits. At the reception the checksum is again computed locally calculated checksum are matched if both are same there is no error else there is an error.

In bit stuffing, each frame begins with a special binary code 01111110 called as flag when error data encounter 5 consecutive ones in the data ,automatically 6^{th} bit is forcibly made 0 at the receiver it sees that 5 ones with o then automatically de stuff or it will detect the 0 to get 1.

Example 1:

Original Data: 0111000111111010

Stuffed data: **01111110**0111000111110**1**0101**01111110**

Algorithm for bit stuffing:

1. Input the data bit stream
2. Check for 5 consecutive ones
3. If present add 0 to the sixth bit
4. If not continue to further bit
5. Add start and end frame
6. Transmit and display

Program

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
int main()
{
char ch,arr[50]={"01111110"},rec[50];
int i,j,k,len=8,cnt=0;
//clrscr();
printf("\n enter the data:\n");
while((ch=getche())!='\r')
{
if(ch=='1')
cnt++;
else
cnt=0;
arr[len++]=ch;
```

```
if(cnt==5)
{
arr[len++]='0';
cnt=0;
}
}
strcat(arr,"01111110");
printf("\n bit stuffed stream is:\n ");
for(i=0;i<len+8;i++)
printf("%c",arr[i]);
//destuffing
cnt=0;
printf("\n the destuffed stream is: \n");
for(j=8,k=0;j<len;j++)
{
if(arr[j]=='1')
cnt++;
else
cnt=0;
rec[k++]=arr[j];
if(cnt==5&&arr[j+1]=='0')
{
j++;
cnt=0;
}
}
for(j=0;j<k;j++)
printf("%c",rec[j]);
getch();
}
```

Output

Original Data: 0111000111111010

Stuffed data: **01111110**01110001111**0**1010**01111110**

Experiment 2: Simulate character stuffing and de-stuffing Using HDLC

Aim: To write and execute program for character stuffing and de-stuffing.

Theory: In character stuffing, at the starting an ASCII sequence **DLESTX** and at the end **DLEETX** is transmitted for synchronization But when DLE occurs in the data itself then the systems adds one more DLE alone with present one so at the receiver it detects one DLE such that DLE of data will be safe

Example:

Original data RAMA

Stuffed data: **DLESTX** RAMA **DLEETX**

Original data DLE RAMADLE

Stuffed data: **DLESTX DLE**DLERAMADLE **DLE DLEETX**

Program:

```
#include<stdio.h>
#include<conio.h>
#define DLE 16
#define STX 2
#define ETX 3
int main()
{
char ch,arr[50]={DLE,STX},rec[50];
int len=2,i,j;
//clrscr();
printf("enter the data stream:ctrl+p->DLE ctrl+b->STX ctrl+c->ETX \n");
while((ch=getch())!='\r')
{
if(ch==DLE)
{
arr[len++]=DLE;
printf("DLE");
}
else if(ch==STX)
printf("STX");
else if(ch==ETX)
printf("ETX");
else printf("%c",ch);
arr[len++]=ch;
}
arr[len++]=DLE;
arr[len++]=ETX;
printf("\n the stuffed stream is:\n");
for(i=0;i<len;i++)
{
ch=arr[i];
if(ch==DLE)
printf("DLE");
```

```
else if(ch==STX)
printf("STX");
else if(ch==ETX)
printf("ETX");
else printf("%c",ch);
}
//destuffing
printf("\n the destuffed data dtream is:\n");
for(j=2;j<len-2;j++)
{
ch=arr[j];
if(ch==DLE)
{
printf("DLE");
j++;
}
else if(ch==STX)
printf("STX");
else if(ch==ETX)
printf("ETX");
else printf("%c",ch);
}
system("pause");
getch ();
return 0;
}
```

Output:

Original data RAMA

Stuffed data: **DLESTX** RAMA **DLEETX**

Original data DLE RAMADLE

Stuffed data: **DLESTX DLE**DLERAMADLE **DLE DLEETX**

Experiment 3: Perform the Encryption and decryption of a given message using substitution method.

Aim: To write and execute a program for encryption and decryption by Using substitution method

Theory: Cryptography (the branch of cryptology dealing with the design of algorithm for encryption and decryption intended entirely for the secrecy and /or authentically of messages) is used for network data protection, privacy and security. In which the message to be encrypted known as plain text are transformed by a function that is a Parameterized by a key. The output of encryption process known as cipher text is then transmitted.

In substitution method each letter or group of letters is replaced by another letter or group of letters to disguise it. The cipher text alphabets to be shifted by K letters. In this case K becomes a key to the generous method of circularly shifted alphabets.

Example:

1. Plain text**:** RAMA

   Key=2

   Cipher text: TCOC

2. Plain text: XYZ

   Key=3

   Cipher text: ABC

Algorithm:

1. Get the message to be encrypted into plain text
2. Initialize the number of shifts per character for encryption
3. Remove non alphabetical character from the plain text and Capitalize them
4. Replace each character of the plain text by shifting letters along the forward direction and put the encrypted message to the cipher text.
5. For decrypting initialize the key K
6. Replace each character of cipher text by shifting k letters along the reverse direction and put the decrypted message to plain text..

Program:

```
#include<stdio.h>
#include<ctype.h>
#include<conio.h>
#define MAX 1000
main()
{
    int shift,pi,ci;
    char plain[MAX],cipher[MAX];
    printf("***encryption and decryption using substitution cipher***\n\n");
    printf("enter the plain text:\n");
    gets(plain);
    while(1)
    {
        printf("\n key (number of shifts per charecter)for encryption:");
        scanf("%d",&shift);
        if(shift<1||shift>25)
        printf("bad input!enter a value between 1 and 25.");
        else
        break;
        }
        printf("\n after removing non alphabetical charecters and capitalizing:\n");
        for(ci=0,pi=0;plain[pi]!='\0';pi++)
        if(isalpha(plain[pi]))
        {
                    putchar(toupper(plain[pi]));
                    cipher[ci++]=((toupper(plain[pi])-'A')+shift%26)%26+'A';
                    }
                    cipher[ci]='\0';
                    printf("\n\n after encryption :\n%s\n",cipher);
                    while(1)
                    {
                        printf("\n key for decryption:");
                        scanf("%d",&shift);
                        if(shift<1||shift>25)
                        printf("bad input!enter a value between 1 and 25.");
                        else
                        break;
                        }
                        for(pi=0,ci=0;cipher[ci]!='\0';ci++)
                        plain[pi++]=((cipher[ci]-'A')+(26-shift))%26+'A';
                        plain[pi]='\0';
                        printf("\n after decryption :\n%s",plain);
                        getch();
                        }
```

Output:

Plain text**:** RAMA

Key=2

Cipher text: TCOC

Plain text: XYZ

Key=3

Cipher text: ABC

Experiment 4. To write a C program to exhibit the working of RSA algorithm.

Algorithm:

1. Two prime numbers are selected as p and q
2. n = pq which is the modulus of both the keys.
3. Calculate totient = (p-1)(q-1)
4. Choose e such that e > 1 and coprime to totient which means gcd (e, totient) must be equal to 1, e is the public key
5. Choose d such that it satisfies the equation de = 1 + k (totient), d is the private key not known to everyone.
6. Cipher text is calculated using the equation c = m^e mod n where m is the message.
7. With the help of c and d we decrypt message using equation m = c^d mod n where d is the private key.

Code:

```
#include<stdio.h>
#include<math.h>
//to find gcd
int gcd(int a, int h)
{
   int temp;
   while(1)
   {
      temp = a%h;
      if(temp==0)
      return h;
      a = h;
      h = temp;
   }
}

int main()
{
   //2 random prime numbers
   double p = 3;
   double q = 7;
   double n=p*q;
   double count;
   double totient = (p-1)*(q-1);

   //public key
   //e stands for encrypt
   double e=2;

   //for checking co-prime which satisfies e>1
   while(e<totient){
   count = gcd(e,totient);
   if(count==1)
      break;
   else
```

```
        e++;
    }

    //private key
    //d stands for decrypt
    double d;

    //k can be any arbitrary value
    double k = 2;

    //choosing d such that it satisfies d*e = 1 + k * totient
    d = (1 + (k*totient))/e;
    double msg = 12;
    double c = pow(msg,e);
    double m = pow(c,d);
    c=fmod(c,n);
    m=fmod(m,n);

    printf("Message data = %lf",msg);
    printf("\np = %lf",p);
    printf("\nq = %lf",q);
    printf("\nn = pq = %lf",n);
    printf("\ntotient = %lf",totient);
    printf("\ne = %lf",e);
    printf("\nd = %lf",d);
    printf("\nEncrypted data = %lf",c);
    printf("\nOriginal Message Sent = %lf",m);

    return 0;
}
```

OUTPUT:

```
Message data = 12.000000
p = 3.000000
q = 7.000000
n = pq = 21.000000
totient = 12.000000
e = 5.000000
d = 5.000000
Encrypted data = 3.000000
Original Message Sent = 12.000000
--------------------------------
Process exited after 8.337 seconds with return value 0
Press any key to continue . . .
```

Congestion Control Using Leaky Bucket Algorithm

The main concept of the leaky bucket algorithm is that the output data flow remains constant despite the variant input traffic, such as the water flow in a bucket with a small hole at the bottom. In case the bucket contains water (or packets) then the output flow follows a constant rate, while if the bucket is full any additional load will be lost because of spillover. In a similar way if the bucket is empty the output will be zero. From network perspective, leaky bucket consists of a finite queue (bucket) where all the incoming packets are stored in case there is space in the queue, otherwise the packets are discarded. In order to regulate the output flow, leaky bucket transmits one packet from the queue in a fixed time (e.g. at every clock tick). In the following figure we can notice the main rationale of leaky bucket algorithm, for both the two approaches (e.g. leaky bucket with water (a) and with packets (b)).

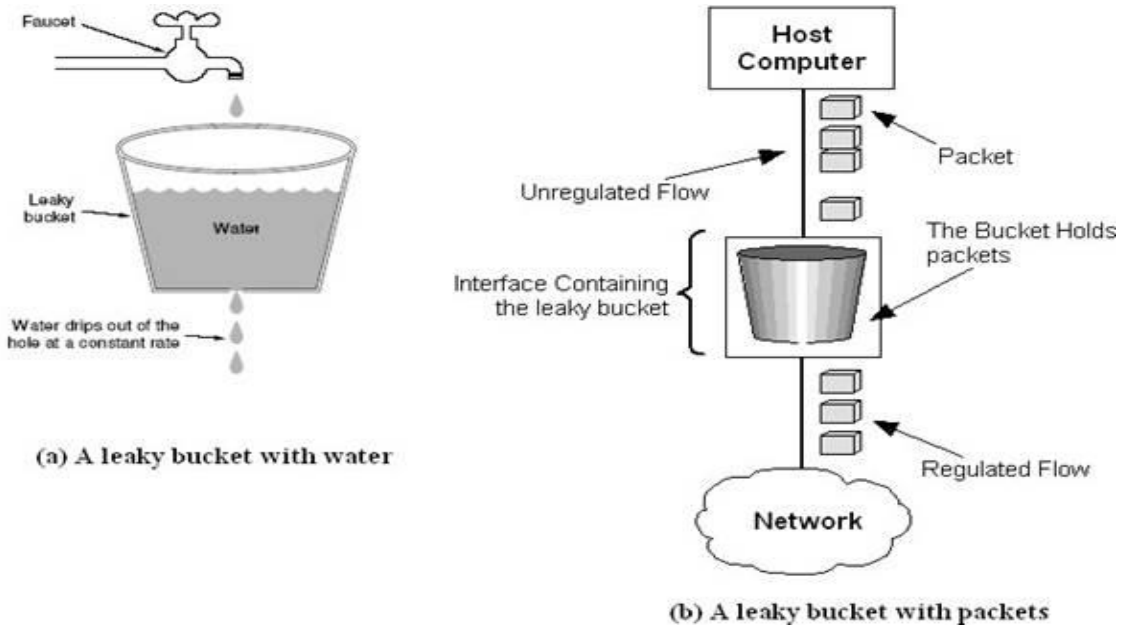


Fig: The leaky bucket traffic shaping algorithm

While leaky bucket eliminates completely bursty traffic by regulating the incoming data flow its main drawback is that it drops packets if the bucket is full. Also, it doesn't take into account the idle process of the sender which means that if the host doesn't transmit data for some time the bucket becomes empty without permitting the transmission of any packet.

Experiment 5. C Program for Congestion control using Leaky Bucket Algorithm

```
#include<stdio.h>
#define min(x,y) ((x) < (y) ? (x :(y))
#define MAX 25

int main()
{
     int cap, oprt, nsec, cont, i=0,dr=0, inp[MAX]={0}, ch, x;
printf("\n\nLEAKY BUCKET ALGORITHM\n");
printf("\nEnter bucket size :   ");
scanf("%d",&cap);
printf("\nEnter output rate (no..of pkts/sec):  ");
scanf("%d",&oprt);
     do{
          printf("\nEnter the no..of packets entering at second       %d:
",i+1);
          scanf("%d",&inp[i++]);
          printf("Enter 1 to insert packets or 0 to quit  :   ");
          scanf("%d",&ch);
       }while(ch);
     nsec=i;
     printf("\nSecond: Packets sent: Packets recvd: In bucket: Dropped\n");
     for(cont=i=0; cont || (i<nsec); i++)
     {
printf(" (%d)   :",i+1);
printf("     (%d)\t: ",/*nsec?*/inp[i]/*:0*/);
printf("   (%d)\t: ",min(cont+inp[i],oprt));
if((x=cont+inp[i]-oprt)>0)
if(x>cap)
{
cont=cap;
dr=x-cap;
}
else
{
cont=x;
dr=0;
}
else
cont=0;
printf("   %d\t:",cont);
printf("     %d\n",dr);
      }
return 0;
}
```

Output:

Leaky bucket algorithm:
Enter bucket size: 3
Enter the output rate (No. of packets/sec): 2
Enter the no. of packets sent at second 1: 6
Enter 1 to insert packets or 0 to quit: 0

Second :	packet sent :	packet recvd :	In bucket :	Dropped
(1)	(6)	(2)	(3)	(1)
(2)	(0)	(2)	(1)	(0)
(3)	(0)	(1)	(0)	(0)

Experiment 6: Simulate the Shortest Path Algorithm

Aim: To stimulate shortest path resulting Algorithm.

Theory: Parting is a major component at N/W layer. This algorithm's key is to determine the shortest path for packet according to cost criterion. Many matrices can be used to assign to each link depending on which functions and to be optimized such as cost capacity, cost packet delay cast congestion etc. In this algorithm the dist. criterion is considered, the source & destination is specified and algorithms fund the node through which the destination can be reached. The shortest distance path through the modes in selected and data is transmitted.

Shortest path is a graph of N/W with each made representing rates and communication links by an are of the graph, labels on are of are computed on a function of dist, band width, cost, delay, avag traffic and other. This program uses dilutra's algorithm to fund the shortest path this algorithm belongs to class of gready algorithms.

DIJKSTRAS ALGORITHM:

1. Enter the distance matrix, get the source node and distinction mode.

2. Set the target node as permanent and present as working node.

3. Examine all the nodes that are adjacent to the current working.

Nodes that are tentative. If the no. of the label on the current working node from the node is smaller than the label on the nodes, we label the node by changing the preceding node entry to current working noted and length entry to the sum of labels on the node & current working node.

4. If the current working node is the source node goto step 5 else check all the nodes labeled tentatively and set the one with smallest label on permanent and current node & goto step or repeat the same.

5. Trace the past from the target node to source node with the node following each node given by the preceding node entry of its label.

Example 1:

Source Node - A

Destination Node-B Total

No. of Nodes – 5

Matrix

	A	B	C	D	E
A	0	0	3	5	6
B	0	0	1	2	7
C	3	1	0	4	0
D	5	2	4	0	0
E	6	7	0	0	0

Program:

```
#include<stdio.h>
#include<conio.h>
 #include<ctype.h>
 #define NON 5
 #define PERM 1
 #define TEMP 0
  struct node
 {
 unsigned int wt;
int prev;
 int state;
     };
 main()
  {
int table[NON][NON]={
{0,0,3,5,6},
{0,0,1,2,7},
{3,1,0,4,0},
 {5,2,4,0,0},
 {6,7,0,0,0},
     };
 int src,dest,workn,i;
   unsigned int min;
struct node nodes[NON];
 //clrscr();
  for(i=0;i<NON;i++)
        {
  nodes[i].state=TEMP;
     nodes[i].wt=-1;
        }
   printf("\n enter source:");
      src=getche();
    workn=src=toupper(src)-'A';
          nodes[src].prev=-1;
         nodes[src].wt=0;
   printf("\n enter destination:");
   dest=toupper(getche())-'A';
          do
           {
   nodes[workn].state=PERM;
    for(i=0;i<NON;i++)
              {
  if(table[workn][i]!=0&&nodes[i].state==TEMP)
              {
    if(nodes[workn].wt+table[workn][i]<nodes[i].wt)
              {
    nodes[i].wt=nodes[workn].wt+table[workn][i];
   nodes[i].prev=workn;
        }
        }
         }
```

```
                min=-1;
  for(i=0;i<NON;i++)
      {
   if(nodes[i].state==TEMP&&nodes[i].wt<min)
         {
min=nodes[i].wt;
         workn=i;
               }
                 }
        }while(workn!=dest);
     printf("\n the shortest path got --> \n %c",dest+65);
          do
        {
   workn=nodes[workn].prev;
     printf("<--%c ",workn+65);
               }
      while(nodes[workn].prev!=-1);
     printf("\n at a total weight of :%d",nodes[dest].wt);
         getch();
                }
```

Part B – (Programs on Networking)

1. Create a three node network topology and connect the duplex links between them. TCL script toobserve the packet flow for the given network in network animator (NAM).

2. Simulate a four node point-to-point network, and connect the links as follows: n0-n2, n1-n2 andn2-n3. Apply TCP agent between n0-n3, n1-n3. Apply relevant applications over TCP agents by changing the parameters and hence determine the number of packets transmitted.

3. Simulate a four node point-to-point network, and connect the links as follows: n0-n2, n1-n2 andn2-n3. Apply UDP agent between n0-n3, n1-n3. Apply relevant applications over UDP agents bychanging the parameters and hence determine the number of packets transmitted.

4. Simulate a three nodes point-to-point network and connect the duplex links between them. Set the queue size, vary the transmission speeds (bandwidth) and find the number of packets dropped. (Point to point network with the links connected as follows: n0 – n2, n1 – n2 and n2 – n3. Apply TCP agent between n0 – n3 and UDP agent between n1 – n3. Apply relevant applications over TCP and UDP agents changing the parameter and determine the number of packets sent by TCP / UDP).

5. Simulate an Ethernet LAN using N-nodes (6-10) with UDP/TCP connection. Apply relevant applications over UDP/TCP agents by changing the parameters and hence determine the number of packets transmitted.

6. Simulate a wireless network for n nodes. For a wireless network consisting of three mobile nodes(n0-n2), Nodes are configured with the specific parameters of a wireless node. Initial location ofthe node is fixed. Nodes are given mobility with fixed speed and fixed destination location. TCPagent is attached to node0 and TCP sink agent is attached to node1. Both the agents are connected and FTP application is attached to TCP agent. Write a TCL script and make an ad-hoc simulation to analyze the output in the trace file. Use the routing protocol as Adhoc on demanddistance vector (AODV).

INTRODUCTION TO SIMULATION TOOLS

WHY NETWORK SIMULATOR?

- Network research area
- Real time Environment
- Single Test Bed: Large amount of time.
- Implementation not easy & Very Costly

INTRODUCTION

- Ns2 is an event driven simulator, which is an open source simulator mainly used for academic research in the areas of Computer Networks, MANETs, WSNs. This provides many possibilities for doing simulation of different protocols before they are actually implemented in real time.
- Network Simulation is a technique where a program models the behavior of a network either by calculating the interaction between the different network entities (hosts/routers, data links, packets, etc.) using mathematical formulas, or actually capturing and playing back observations from a production network.
- When a simulation program is used in conjunction with live applications and services in order to observe end-to-end performance to the user desktop, this technique is also referred to as network emulation.

WHAT'S THE PURPOSE OF NS2?

- To set-up a network topology on your own and analysis the behavior.
- To evaluate or to test a performance of exiting network protocol such as traffic generators (e.g., web, telnet),delay, jitter, packet data, link loss models etc. by modelling the network
- To develop or to prototype a new Network Protocol and evaluate the protocol with support for the most popular protocols in use today, such as IPv4, IPv6, UDP, and TCP.
- Large-scale or virtual environment simulations not possible in real experiment.
- Design different network technology
- Network simulators can also provide other tools to facilitate visual analysis of trends and potential trouble spots.
- Certain simulators have added functionality of capturing this type of data directly from a functioning production environment, at various times of the day, week, or month, in order to reflect average, worst-case, and best-case conditions.
- Most of the commercial simulators are GUI driven, while some network simulators require input scripts or commands (network parameters that describes the state of the network such as node placement, existing links and the events such as data transmissions, link failures, etc.)

- The most popular Open-Source Simulators available in the market are NS (also called NS-2), PDNS (Parallel/Distributed NS), GloMoSim, SSFNet (Scalable Simulation Framework Net Models), DaSSF (Dartmouth SSF), OMNET++ and others.
- The popular Commercial Network Simulators are OpNet Modeler, QualNet, NetWiser, Shunra, NetScale, NetSim and others.

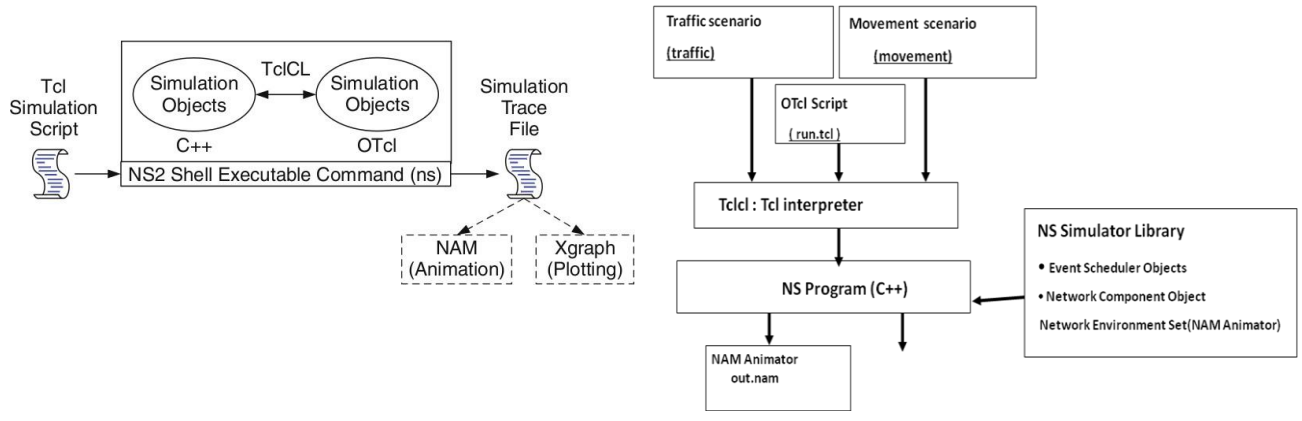
FEATURES OF NS2

- Discrete event scheduler, packet level simulator
- It provides substantial support to simulate set of protocols (TCP, FTP, UDP, HTTP & DSR)
- it simulates wired and wireless network and emulation modes
- It is primarily UNIX based.
- Written in C++ with Otcl front-end (Object oriented support) and uses TCL as its scripting language. Known as Tclcl: C++ and otcl linkage

PLATFORMS

- Most UNIX and UNIX-like systems
  - Linux
  - FreeBSD
  - SunOS/Solaris
  - HP/SGI (with some tweaking)
- Windows 95/98/NT/ME/2000/XP
  - Best based on cygwin
  - Tips on build available
  - However validation tests don't work

NS2 ARCHITECTURE



- Object-oriented (C++ and Otcl)
- oTCL in the Front End(For Control)
  - Control part of NS2
  - Topology (Simulation scenario) configurations, “one-time” task
  - Event driven: Periodic or Triggered action
  - Manipulates existing C++ objects
  - Easy and fast to write and edit
  - Fast to run, quick to re-configure
  - Fine grained object composition, Network topology, Network objects
- C++ in the Back End (For Data)
  - Core of NS2, data part of NS2
  - Easy to modify the code: Not fully layered and structured
  - Packet processing and execution
  - Per packet action
  - Algorithms over large data sets, per packet handling in C++
- It Compromise between compatibility and speed
- Require hard work in Learning and debugging

DESIGN OF NS-2

The NS2 makes use of flat earth model in which it assumes that the environment is flat without any elevations or depressions. However the real world does have geographical features like valleys and mountains. NS2 fails to capture this model in it. So NS3 simulator is used

Creating the Topology

- Decide what do you want to simulate
  - Wired or wireless network
  - What are the protocols?
  - How many nodes, what are the measuring parameters?
  - What are the applications involved, etc?
- Make a rough sketch of the topology
  - Based on the requirement edit the existing C++ files and/or the .tcl files or create new C++ files/.tcl files

Directory Structure

- Main directories
  - bin, ns-2xx, lib, man, include, etc in ns2 home
- ns-2.xx
  - Readme file
  - Makefile, installation file, tutorial, etc
  - Source files related to the protocols
    - All .cpp and .h files related needed for editing
  - Need understanding of interaction among the functions/sub routines
  - Not fully layered like QualNet

Compiling Ns2

- Create / Modify the C++ file
  - If you are creating new C++ file, include the name of the new files in the Make file.
  - If you are editing the existing C++ files, keep a copy of the original file
  - Add comments to your modifications with date
- Compile NS2
  - After creation/editing, compile NS2 using
    - (make clean;) make;
    - Check for errors, if any and rectify

Executing Ns2

- Create your .tcl script as per your topology
- Run the .tcl file using *ns* command
  - Check which ns2 you are using
  - Create a huge output file (trace file) to analyze
  - Need to understand the file contents
  - Perl scripts are also available to analyze the trace file
- Analyze using *nam* Visual network animator
- Single thread of control and No locking or race conditions to worry about.

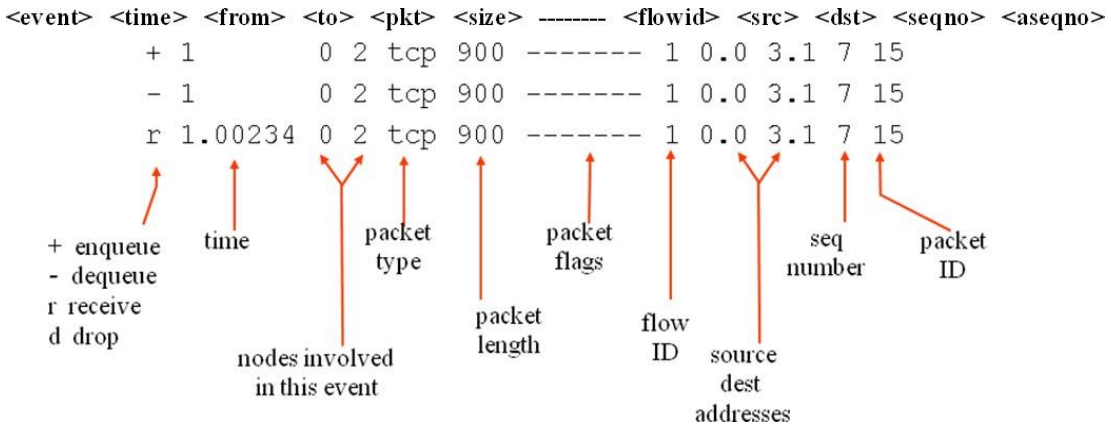
SUPPORTING TOOLS WITH NS2: TRACES AND MONITORS

Trace Files using Otcl Script

- It can document every event that occurred in the simulation and is used for analysis.
- It contains the topological information like nodes, links and packet traces.
- Packet traces describe each individual packet as they travel through the network, transmission of the packet on a link, Packet dropped at a link or queue using tcl script

- The ns2 contains various trace format for suitable use in different simulation such as wireless trace format, AODV Routing algorithm, DSDV routing algorithm etc.

- **Trace file format**



Trace Via Monitor

- Queue monitor: the queue to access a link
- Flow monitor: particular flow within queue

To specify a link, we need to set the link

set link [$ns link $n0 $n1]

To create a flow monitor:

set fmon [$ns makeflowmon Fid]

$ns attach-fmon $link $fmon

$ns at <time> “puts $fmon set pdrops_”

NAM (Network Animator) Files

- NAM is a TCL-based animation tool for viewing network simulation traces and real traces of packet data.
- To use NAM, you have to first generate a trace file
- Once the trace file is generated, NAM will read it, create a topology, pop up a window, do the layout if necessary, and then pause at the time of the first packet in the trace file.
- NAM provides control over many aspects of animation through its user interface, and does animation using the following building blocks: node, link, queue, packet, agent and monitor.

XGRAPH

- It is a plotting program that is used to create graphic representations of simulation results.
- It is important because it allows some basic animation of data sets.
- The animation only pages through data sets in the order in which they are loaded.
- It is quite crude, but useful if all the data sets are in one file in the time order, and are put out at uniform intervals.

– Also, the code will take derivatives of your data numerically and display these in a new XGraph window.

BASICS OF TCL

Syntax: command arg1 arg2 arg3
Hello World!

```
puts stdout{Hello, World!}
Hello, World!
```

Variables Command Substitution

```
set a 5          set len [string length foobar]

set b $a         set len [expr [string length foobar] + 9]
```

Simple Arithmetic

```
expr 7.2 / 4
```

Procedures

```
proc Diag {a b} {

set c [expr sqrt($a * $a + $b * $b)]

return $c }

puts —Diagonal of a 3, 4 right triangle is [Diag 3 4]
```

Output: Diagonal of a 3, 4 right triangle is 5.0

Loops

```
while{$i < $n} {

        . . . . . .

}

for {set i 0} {$i < $n} {incr i} {

}
```

Wired TCL Script Components

- Create the event scheduler
- Open new files & turn on the tracing
- Create the nodes

- Setup the links
- Configure the traffic type (e.g., TCP, UDP, etc)
- Set the time of traffic generation (e.g., CBR, FTP)
- Terminate the simulation

NS Simulator Preliminaries.

1. Initialization and termination aspects of the ns simulator.
2. Definition of network nodes, links, queues and topology.
3. Definition of agents and of applications.
4. The NAM visualization tool.
5. Tracing and random variables.

Initialization and Termination of TCL Script in NS-2

An ns simulation starts with the command

```
set ns [new Simulator]
```

Which is thus the first line in the tcl script? This line declares a new variable as using the set command, you can call this variable as you wish, In general people declares it as ns because it is an instance of the Simulator class, so an object the code[new Simulator] is indeed the installation of the class Simulator using the reserved word new.

In order to have output files with data on the simulation (trace files) or files used for visualization (nam files), we need to create the files using —open command:

#Open the Trace file

```
set tracefile1 [open out.tr w]
$ns trace-all $tracefile1
```

#Open the NAM trace file

```
set namfile [open out.nam w]
$ns namtrace-all $namfile
```

The above creates a data trace file called —out.tr and a nam visualization trace file called —out.nam .Within the tcl script, these files are not called explicitly by their names, but instead by pointers that

are declared above and called —tracefile1 and —namfile respectively. Remark that they begins with a # symbol. The second line open the file —out.tr to be used for writing, declared with the letter —w. The third line uses a simulator method called trace-all that have as parameter the name of the file where the traces will go.

The last line tells the simulator to record all simulation traces in NAM input format .It also gives the file name that the trace will be written to later by the command $ns flush-trace. In our case, this will be the file pointed at by the pointer —$namfile ,i.e the file —out.tr .

The termination of the program is done using a —finish procedure.

#Define a finish procedure

```
Proc finish { } {
global ns tracefile1 namfile
$ns flush-trace
Close $tracefile1
Close $namfile
Exec nam out.nam &
Exit 0
}
```

The word proc declares a procedure in this case called finish and without arguments. The word global is used to tell that we are using variables declared outside the procedure. The simulator method —flush-trace" will dump the traces on the respective files. The tcl command —close" closes the trace files defined before and exec executes the nam program for visualization. The command exit will ends the application and return the number 0 as status to the system. Zero is the default for a clean exit. Other values can be used to say that is a exit because something fails.

At the end of ns program we should call the procedure —finish and specify at what time the termination should occur. For example,

```
$ns at 125.0 "finish"
```

Will be used to call —finish at time 125sec.Indeed, the at method of the simulator allows us to schedule events explicitly.

The simulation can then begin using the command

```
$ns run
```

Definition of a network of links and nodes

The way to define a node is

```
set n0 [$ns node]
```

The node is created which is printed by the variable n0. When we shall refer to that node in the script we shall thus write $n0.

Once we define several nodes, we can define the links that connect them. An example of a definition of a link is:

```
$ns duplex-link $n0 $n2 10Mb 10ms DropTail
```

Which means that $n0 and $n2 are connected using a bi-directional link that has 10ms of propagation delay and a capacity of 10Mb per sec for each direction.

To define a directional link instead of a bi-directional one, we should replace —duplex-link by —simplex-link.

In NS, an output queue of a node is implemented as a part of each link whose input is that node. The definition of the link then includes the way to handle overflow at that queue. In our case, if the buffer capacity of the output queue is exceeded then the last packet to arrive is dropped. Many alternative options exist, such as the RED (Random Early Discard) mechanism, the FQ (Fair Queuing), the DRR (Deficit Round Robin), the stochastic Fair Queuing (SFQ) and the CBQ (which including a priority and a round-robin scheduler).

In ns, an output queue of a node is implemented as a part of each link whose input is that node. We should also define the buffer capacity of the queue related to each link. An example would be:

```
#set Queue Size of link (n0-n2) to 20

$ns queue-limit $n0 $n2 20
```

Agents and Applications

We need to define routing (sources, destinations) the agents (protocols) the application that use them.

FTP over TCP

TCP is a dynamic reliable congestion control protocol. It uses Acknowledgements created by the destination to know whether packets are well received.

There are number variants of the TCP protocol, such as Tahoe, Reno, New Reno, Vegas. The type of agent appears in the first line:

```
set tcp [new Agent/TCP]
```

The command $ns attach-agent $n0 $tcp defines the source node of the tcp connection.

The command

```
set sink [new Agent /TCPSink]
```

Defines the behavior of the destination node of TCP and assigns to it a pointer called sink.

#Setup a UDP connection

```
set udp [new Agent/UDP]
$ns attach-agent $n1 $udp
set null [new Agent/Null]
$ns attach-agent $n5 $null
$ns connect $udp $null
$udp set fid_2
```

#setup a CBR over UDP connection

The below shows the definition of a CBR application using a UDP agent

The command $ns attach-agent $n4 $sink defines the destination node. The command $ns connect $tcp $sink finally makes the TCP connection between the source and destination nodes.

```
set cbr [new
Application/Traffic/CBR]
$cbr attach-agent $udp
$cbr set packetsize_ 100
$cbr set rate_ 0.01Mb
$cbr set random_ false
```

TCP has many parameters with initial fixed defaults values that can be changed if mentioned explicitly. For example, the default TCP packet size has a size of 1000bytes.This can be changed to another value, say 552bytes, using the command $tcp set packet Size_ 552.

When we have several flows, we may wish to distinguish them so that we can identify them with different colors in the visualization part. This is done by the command $tcp set fid_ 1 that assigns to the TCP connection a flow identification of —1 .We shall later give the flow identification of —2 to the UDP connection.

CBR over UDP

A UDP source and destination is defined in a similar way as in the case of TCP.

Instead of defining the rate in the command $cbr set rate_ 0.01Mb, one can define the time interval between transmission of packets using the command.

```
$cbr set interval_ 0.005
```

The packet size can be set to some value using

```
$cbr set packetSize_ <packet size>
```

Scheduling Events

NS is a discrete event based simulation. The tcp script defines when event should occur. The initializing command set ns [new Simulator] creates an event scheduler, and events are then scheduled using the format:

```
$ns at <time> <event>
```

The scheduler is started when running ns that is through the command $ns run.

The beginning and end of the FTP and CBR application can be done through the following command

```
$ns at 0.1 "$cbr start"
$ns at 1.0 " $ftp start"
$ns at 124.0 "$ftp stop"
$ns at 124.5 "$cbr stop"
```

Structure of Trace Files

When tracing into an output ASCII file, the trace is organized in 12 fields as follows in fig shown below, The meaning of the fields are:

Event	Time	From Node	To Node	PKT Type	PKT Size	Flags	Fid	Src Addr	Dest Addr	Seq Num	Pkt id

1. The first field is the event type. It is given by one of four possible symbols r, +, -, d which correspond respectively to receive (at the output of the link), enqueued, dequeued and dropped.
2. The second field gives the time at which the event occurs.
3. Gives the input node of the link at which the event occurs.
4. Gives the output node of the link at which the event occurs.
5. Gives the packet type (eg CBR or TCP)
6. Gives the packet size
7. Some flags
8. This is the flow id (fid) of IPv6 that a user can set for each flow at the input OTcl script one can further use this field for analysis purposes; it is also used when specifying stream color for the NAM display.
9. This is the source address given in the form of "node.port".
10. This is the destination address, given in the same form.
11. This is the network layer protocol's packet sequence number. Even though UDP implementations in a real network do not use sequence number, ns keeps track of UDP packet sequence number for analysis purposes
12. The last field shows the Unique id of the packet.

XGRAPH

The Xgraph program draws a graph on an x-display given data read from either data file or from standard input if no files are specified. It can display up to 64 independent data sets using different colors and line styles for each set. It annotates the graph with a title, axis labels, grid lines or tick marks, grid labels and a legend.

```
Xgraph [options] file-name
```

Options are listed here

/-bd <color> (Border)

This specifies the border color of the xgraph window.

/-bg <color> (Background)

This specifies the background color of the xgraph window.

/-fg<color> (Foreground)

This specifies the foreground color of the xgraph window.

/-lf <fontname> (LabelFont)

All axis labels and grid labels are drawn using this font.

/-t<string> (Title Text)

This string is centered at the top of the graph.

/-x <unit name> (XunitText)

This is the unit name for the x-axis. Its default is "X".

/-y <unit name> (YunitText)

This is the unit name for the y-axis. Its default is "Y".

HOW TO CREATE AN OTCL SCRIPT FOR YOUR NETWORK MODEL

CREATING NETWORK TOPOLOGY

To be able to run a simulation scenario, a network topology must first be created. In ns2, the topology consists of a collection of nodes and links.

Before the topology can be set up, a new simulator object must be created at the beginning of the script with the command:

 # .....Create an Event scheduler object creation .....#

set ns [new Simulator]

 # .....Event scheduler event.....#

$ns at <time> <event>

Event is an object in C++ hierarchy, With unique ID, Scheduled time, pointer to an object. Scheduler maintains the ordered data structure with the events to be executed and fires them one by one, invoking the handler of the event.

The simulator object has member functions that enable creating the nodes and the links, connecting agents etc. All these basic functions can be found from the class Simulator. When using functions belonging to this class, the command begins with "$ns", since ns was defined to be a handle to the Simulator object.

Steps:

- Create nodes required for the network model
- Create Links that connect together two nodes (Simplex, half duplex and duplex links)
- Assign positions to nodes in the network (with infrastructure)
- Additional Instructions For Visualization of Network wrt Setting color ID, Labelling the nodes, Colouring the nodes, geometrical shapes to the nodes in network etc

New node objects can be created with the command:

 # .....Creating nodes required for the network....# **set <nodes> [$ns node]**

set n0 [$ns node]

set n1 [$ns node]

set n2 [$ns node]

set n3 [$ns node]

(Alternate way)

set totalNodes <no of nodes required>

for {set i 0} {$i < $totalNodes} {incr i} {

set node_($i) [$ns node]

}

New Links objects can be created with the command:

 # .....Creating Duplex Link. ..... #

$ns duplex-link $<node> $<node> <bandwidth> <delay> <queue_type>

The member function of the Simulator class, called "node" creates four nodes and assigns them to the handles n0, n1, n2 and n3. These handles can later be used when referring to the nodes. If the node is not a router but an end system, traffic agents (TCP, UDP etc.) and traffic sources (FTP,CBR etc.) must be set up, i.e, sources need to be attached to the agents and the agents to the nodes, respectively.

Assign positions to nodes in the network (with infrastructure) can be done with the command:

#..... Give node position (for NAM) ie, Orientation to duplex links and nodes ..... #

$ns duplex-link-op $<node> $<node> < position as orient right, orient right-up etc.

Visualization of Network can be done with the command:

#.....Setting color ID..... #

$ns color 1 <choose the color>

#.....Labelling the nodes ..... #

$ns at 0.0 "$ <nodes> label <labelname>"

#.....Coloring the nodes ..... #

$ns at 0.0 "$ <nodes>color <choose color such as yellow,blue, green etc.>"

#.....Assigning the geometrical shapes to the nodes in network. ..... #

$n0 shape <choose the shape such as hexagon, circle etc.>

Parameterize Simulation Objects such as Duplex links parameters are BW(data rate/bitrate determines the rate of information), delay(determines propagation delay), queue type(determines queueing policy such as DropTail, RED, CBQ, FQ, SFQ, DRR), Links: queue sizes, link speeds, datadate. No of packets etc. can be done with the command:

 # .....Duplex Link Parameters ..... #

$ns duplex-link $<node> $<node> <bandwidth> <delay> <queue_type>

#.....Set Queue Size of link. ..... #

$ns queue-limit $<node> $<node> <queue size>

#.....Monitor the queue for link for NAM. ..... #

$ns duplex-link-op $<node> $<node> < queuePos 0.5>

CREATE CONNECTIONS

Agents, applications and traffic sources

- Transports:TCP, UDP, multicast, etc.
- Create the TCP Connection source
- Attach the Agent with Node
- Connection of source with the Sink
- Similarily with UDP Connection also.

#.....Setup of a TCP connection ..... #
Set tcp [new Agent/TCP]
$ns attach-agent $<nodes> $tcp
#.....Setup aTCPsink connection ..... #
Set sink [new Agent/TCPSink]
$ns attach-agent $<node> $sink
#.....TCP connection with sink. ..... #
$ns connect $tcp $sink
#.....Setup of a UDP connection ..... #
Set udp [new Agent/UDP]
$ns attach-agent $<nodes> $udp

#.....Setup a UDP Null connection ..... #
Set null [new Agent/Null]
$ns attach-agent $n3 $null
#.....UDP connection with Null ..... #
$ns connect $udp $null

CREATE TRAFFIC MODELS IE, CREATING TRAFFIC OVER TCP/UDP

- Traffic (such as Web, ftp, telnet, audio, etc)
- Setup a FTP over TCPconnection : default is “infinite” file size
- Application objects attach to transport protocol objects
- Generates traffic into transport protocol
- Schedule events for the FTP agents
- Similarily with UDP Channels also
- Displaying the parameter for anaysis of the network
- Detach tcp and sink agents (not really necessary)

#.....Create Traffic Model How to create a FTP session over TCP?.....
#.....Setup a FTP over TCP connection.....#
Set ftp [new Application/FTP]
#.....Application objects attach to transport protocol objects ie, TCP Channel .....#
$ftp attach-agent $tcp
#.....Schedule events for the FTP agents. ..... #
$ns at <time=1.0> "$ftp start"$ns at <time=4.0> "$ftp stop"

#.....Create Traffic Model How to create a CBR (Constant Bit Rate) model over UDP? .....#
#.....Setup a CBR over UDP connection .....#
Set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
$cbr set type_ <CBR or VBR>
$cbr set packet_size_ <No of packets>
$cbr set rate_ <cbr rate>
$cbr set random_ <cbr mode>
#.....Schedule events for the CBR agents. ..... #
$ns at <time=0.1> "$cbr start"
$ns at <Time = 4.5 >"$cbr stop"

#.....Detach tcp and sink agents (not really necessary).....#
$ns at <timeduration=4.5> "$ns detach-agent $n0 $tcp ; $ns detach-agent $n3 $sink"

#.....Displaying the parameter for anaysis of the network.....#
#..... Print CBR packet size and interval.....#
Puts "CBR packet size = [$cbr set packet_size_]"
puts "CBR interval = [$cbr set interval_]"

COLLECT STATISTICS: ENABLE TRACING & NAM

- dump everything to trace, post process it
- gather stats during simulation within OTCL script
- modify Ns source code

```
#.............Creating trace objects ..... #
Set nt [open out.tr w]
$ns trace-all $nt

#.........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
}

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

RUN NS MULTIPLE TIMES

```
# ..... Start scheduler. ..... #
$ns run
```

NS2 EXECUTION

.TCL files contain the code of network simulation and can be executed by following the steps given below:

1. Once you have installed the NS2 successfully, open a terminal and run the following:
   % ns <filename.tcl>

2. Once you run that command, .tr and .nam files will be created in the same directory that contains the .TCL file
3. To interpret the results in the form of animation, you need to just run the command
   % nam <filename.nam >

4. And to see the results in numerical form, you need to open the .tr file in the editor using command
   % gedit <filename.tr>

BASIC LINUX COMMANDS THAT ARE NEEDED FOR NS2

- ls : listing the file contents
- cd: change the directory, Returns you to your login directory
- cd ~ : Also returns you to your login directory

- cd /home : Takes you to the home directory, where user login directories are usually stored
- mkdir : make a directory
- gedit : it is similar like a notepad editor in windows(syntax is gedit filename)
- vi : it is vi editor
- clear: clears the terminal
- pwd : it is just present working directory
- ./install : (See a dot in the beginning, it is just executing the install file which is available under a folder, any executable can be executed with the help of this ./)
- chmod : Changing the mode of the file (syntax is chmod 755 <filename>
- echo : it echo the value to the screen
    - Example: echo name

  echo $SHELL (it will print the name of the shell)

  echo $USER (it will print the name of the user)

  echo $PATH (it display the values contained in the PATH variable)

  echo $HOME (it displays the home folder of the user) (Now you can understood that the variable names are denoted using a $ symbol)

INSTALLATION STEPS

1. Open the terminal and change the working directory to where you downloaded the ns-allinone package.
2. Now untar (uncompress) the package using the following command
   tar xyz filename
3. Change the directory to *ns-allinone2.29*, run the installation script using the following command and wait until the installation is successfully completed
   ./install
4. After successfully installing the NS2 package, configure the *.bashrc* file, wINSTAhich is present in the following path
   home/ns2username
5. Edit the *.bashrc* file using the following command
   gedit .bashrc
6. Set the following path in the last line of *.bash*
   # export PATH="$PATH:/home/ns2userName/ns-allinone-2.35/bin:/ home /ns2userName/ns-allinone-2.35/tcl 8.4.11/unix:/home/ ns2userName/ns-allinone-2.35/tk8.4.11/unix" export LD_LIBRARY_PATH="/home/ns2userName/ns-allinone-2.35/ otcl-1.11,/home/ns2userName/ns-alli none-2.35/lib" export TCL_ LIBRARY="/home/ns2userName/ns-allinone-2.35/ tcl8.4.11/library"
7. If the path set is correct, open the new terminal and run the following in the home directory: ns/

8. If you see a % sign displayed in the terminal, congratulations, you have installed NS2 successfully!
9. Now to validate the installation, run the following command (optional): cd/ns-allinone-2.35/ns-2.35 / ./validate

Path setting is important:

The path can be set in a file which is available for each user of the Linux OS. For example, a user called "pradeep" will have the home folder as /home/pradeep and another user "kumar" has home folder as /home/kumar

Each user "pradeep" and "kumar" will have a file called .bash_profile (in case of redhat and fedora) and .bashrc (Ubuntu, etc) in their home folder. This file is a hidden file which can be opened using a vi or gedit editor

The command being

vi /home/pradeep/.bash_profile

vi /home/pradeep/.bashrc

Or

gedit /home/pradeep/.bash_profile

gedit /home/pradeep/.bashrc

Similarly the same for user "kumar" (if you set the path for "pradeep" and running under "kumar", it will not work, for that to work, there is a separate customisation is available), for beginners, these steps are enough

GETTING STARTED

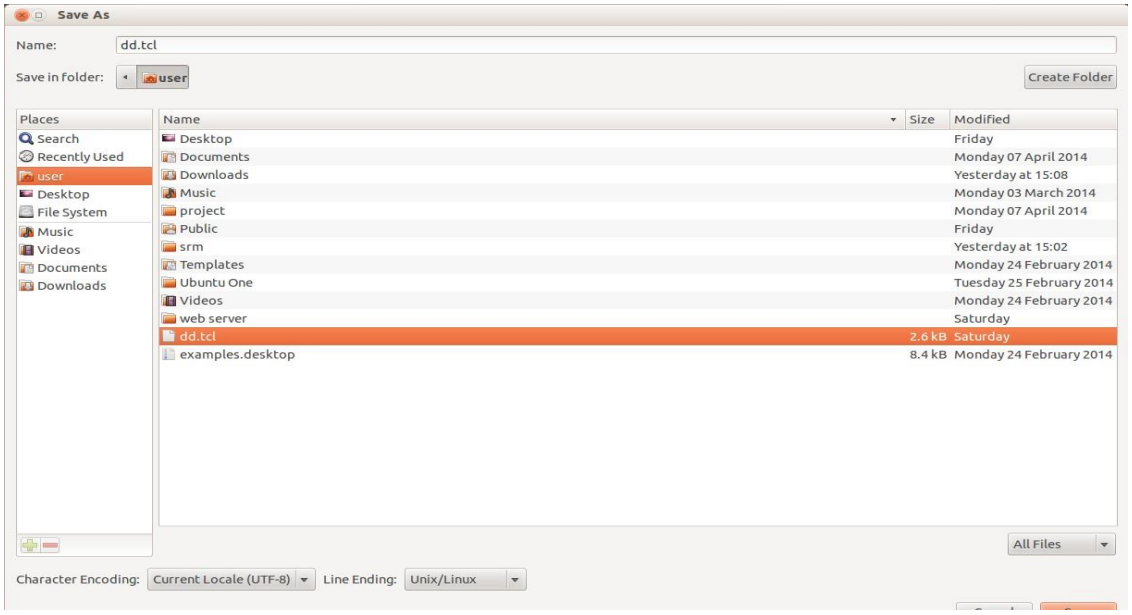
HOW TO RUN TCL PROGRAMS

1. **Open gedit**

   Application --> Accessories --> gedit Text editor

2. **Write your tcl script on gedit text editor**

3. **Save file with .tcl extension**



After writing program, save it in .tcl format. Save your file with .tcl extension.

4. **After that you can see your program changed like this i.e as multicolored**

```
Open  Save  Undo
*dd.tcl
#####################################################################
#################http://www.ns2blogger.blogspot.in/##################
#####################################################################

set ns [new Simulator]
set node_(s1) [$ns node]
set node_(s2) [$ns node]
set node_(r1) [$ns node]
set node_(r2) [$ns node]
set node_(s3) [$ns node]
set node_(s4) [$ns node]

$ns duplex-link $node_(s1) $node_(r1) 10Mb 2ms DropTail
$ns duplex-link $node_(s2) $node_(r1) 10Mb 3ms DropTail
$ns duplex-link $node_(r1) $node_(r2) 1.5Mb 20ms RED
$ns queue-limit $node_(r1) $node_(r2) 25
$ns queue-limit $node_(r2) $node_(r1) 25
$ns duplex-link $node_(s3) $node_(r2) 10Mb 4ms DropTail
$ns duplex-link $node_(s4) $node_(r2) 10Mb 5ms DropTail

$ns duplex-link-op $node_(s1) $node_(r1) orient right-down
$ns duplex-link-op $node_(s2) $node_(r1) orient right-up
$ns duplex-link-op $node_(r1) $node_(r2) orient right
$ns duplex-link-op $node_(r1) $node_(r2) queuePos 0
$ns duplex-link-op $node_(r2) $node_(r1) queuePos 0
$ns duplex-link-op $node_(s3) $node_(r2) orient left-down
$ns duplex-link-op $node_(s4) $node_(r2) orient left-up


set tcp1 [$ns create-connection TCP/Reno $node_(s1) TCPSink $node_(s3) 0]
$tcp1 set window_ 15
set tcp2 [$ns create-connection TCP/Reno $node_(s2) TCPSink $node_(s3) 1]
$tcp2 set window_ 15
set ftp1 [$tcp1 attach-source FTP]
set ftp2 [$tcp2 attach-source FTP]

# Tracing a queue
set redq [[$ns link $node_(r1) $node_(r2)] queue]
                                    Tcl  Tab Width: 8  Ln 5, Col 1  INS
```

5. **Run the program in Terminal**

To Open terminal Application --> Accessories --> terminal

6. **To run the tcl script you have to change to the directory where the program is located**

Make use of cd

```
user@user-laptop:~$ cd /home/user/project/PROGRAM
user@user-laptop:~/project/PROGRAMS$
```

Run The TCL Script by typing command ns <filename>.tcl

```
user@user-laptop:~$ cd /home/user/project/PROGRAM
user@user-laptop:~/project/PROGRAM$ ns aodv.tcl
```

Program is running

```
user@user-laptop:~/project/PROGRAM/aodv$ ns aodv.tcl
num_nodes is set 30
INITIALIZE THE LIST xListHead
Starting Simulation...
SORTING LISTS ...DONE!
channel.cc:sendUp - Calc highestAntennaZ_ and distCST_
highestAntennaZ_ = 1.5,  distCST_ = 550.0
```

7. **Now simulation started and you can get network animator window, xgraph and trace files**

EXPERIEMENT 1

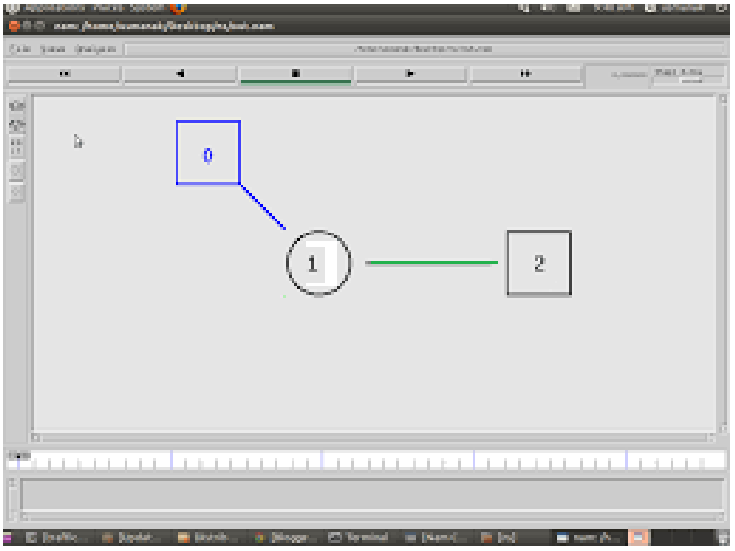
Create a three node network topology and connect the duplex links between them.

Program:

```
set ns [new Simulator]

set tracefile [open lab1.tr w]
$ns trace-all $tracefile

set namfile [open lab1.nam w]
$ns namtrace-all $namfile

proc finish {} {
global ns tracefile namfile
$ns flush-trace
close $tracefile
close $namfile
exec nam lab1.nam &
exit 0
}

set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]

$ns duplex-link $n0  $n1 10.0Mb 1ms DropTail
$ns queue-limit $n0  $n1 10
$ns duplex-link $n1  $n2 10.0Mb 1ms DropTail
$ns queue-limit $n1  $n2 10

set tcp0 [new Agent/TCP]
```

```
$ns attach-agent $n0 $tcp0
set sink2 [new Agent/TCPSink]
$ns attach-agent $n2 $sink2
$ns connect $tcp0 $sink2

set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
$ns at 1.0 "$ftp0 start"
$ns at 10.0 "$ftp0 stop"
$ns at 11.0 "finish"
$ns run
```

SIMULATION RESULTS



Topology

EXPERIEMENT 2

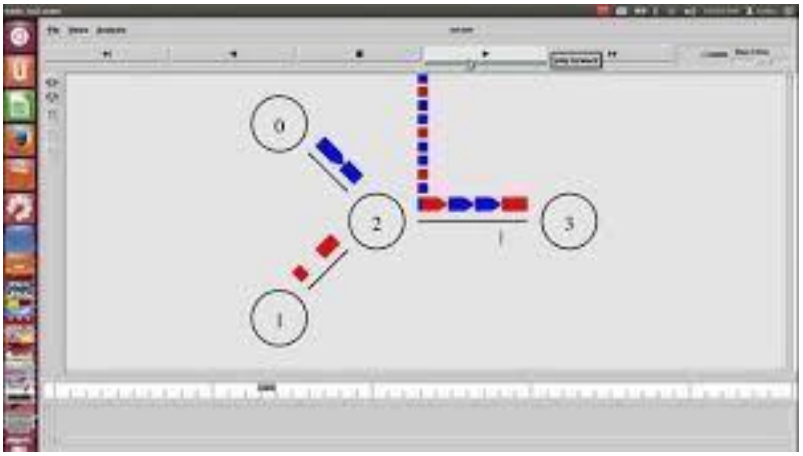
Simulate a four/three node Point-To-Point Network and Connect the Links as Follows: n0-n2, n1-n2 and n2-n3. Apply TCP Agent between n0-n3, n1-n3. Apply Relevant Applications over TCP Agents by changing the Parameters and Hence Determine the Number of Packets transmitted.

Program:

```
set ns [new Simulator]

set tracefile [open 2.tr w]
$ns trace-all $tracefile

set namfile [open 2.nam w]
$ns namtrace-all $namfile

proc finish {} {
global ns tracefile namfile
$ns flush-trace
close $tracefile
close $namfile
exec nam 2.nam &
exit 0

}

set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]

$ns duplex-link $n0 $n2 10.0Mb 1ms DropTail
$ns queue-limit  $n0 $n2 10
$ns duplex-link $n1 $n2 10.0Mb 1ms DropTail
$ns queue-limit  $n1 $n2 10
$ns duplex-link $n2 $n3 10.0Mb 1ms DropTail
$ns queue-limit  $n2 $n3 10

set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0

set sink3 [new Agent/TCPSink]
$ns attach-agent $n3 $sink3
$ns connect $tcp0 $sink3

set tcp1 [new Agent/TCP]
$ns attach-agent $n1 $tcp1

set sink3 [new Agent/TCPSink]
$ns attach-agent $n3 $sink3
$ns connect $tcp1 $sink3

set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
$ns at 1.0 "$ftp0 start"
```

```
$ns at 10.0 "$ftp0 stop"

set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1
$ns at 1.0 "$ftp1 start"
$ns at 10.0 "$ftp1 stop"

$ns at 11.0 "finish"

$ns run
```

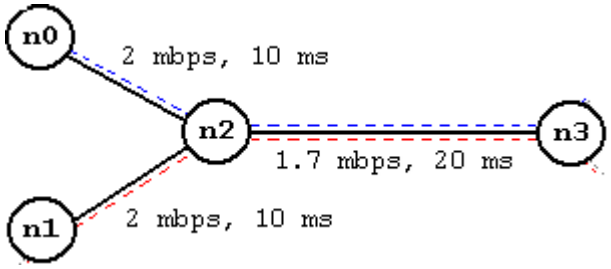
SIMULATION RESULTS



Topology

EXPERIEMENT 3

Simulate a four/three node Point-To-Point Network and Connect the Links as Follows: n0-n2, n1-n2 and n2-n3. Apply UDP Agent between n0-n3, n1-n3. Apply Relevant Applications over TCP Agents by changing the Parameters and Hence Determine the Number of Packets transmitted.

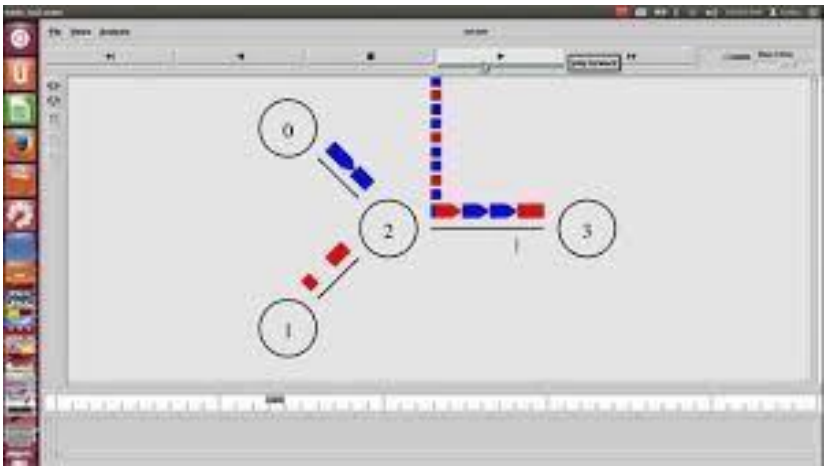


#Create a simulator object

```
set ns [new Simulator]

set tracefile [open 2.tr w]
$ns trace-all $tracefile

set namfile [open 2.nam w]
$ns namtrace-all $namfile

proc finish {} {
global ns tracefile namfile
$ns flush-trace
close $tracefile
close $namfile
exec nam 2.nam &
exit 0
}

set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]

$ns duplex-link $n0 $n2 10.0Mb 1ms DropTail
$ns queue-limit $n0 $n2 10
$ns duplex-link $n1 $n2 10.0Mb 1ms DropTail
$ns queue-limit $n1 $n2 10
$ns duplex-link $n2 $n3 10.0Mb 1ms DropTail
$ns queue-limit $n2 $n3 10

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

set null3 [new Agent/Null]
$ns attach-agent $n3 $null3
$ns connect $udp0 $null3

set udp1 [new Agent/UDP]
$ns attach-agent $n1 $udp1

set null3 [new Agent/Null]
```

```
$ns attach-agent $n3 $null3
$ns connect $udp1 $null3

set cbr0 [new Application/Traffic/CBR]
$cbr0 attach-agent $udp0
$ns at 1.0 "$cbr0 start"
$ns at 10.0 "$cbr0 stop"

set cbr1 [new Application/Traffic/CBR]
$cbr1 attach-agent $udp1
$ns at 1.0 "$cbr1 start"
$ns at 10.0 "$cbr1 stop"

$ns at 11.0 "finish"

$ns run
```

SIMULATION RESULTS

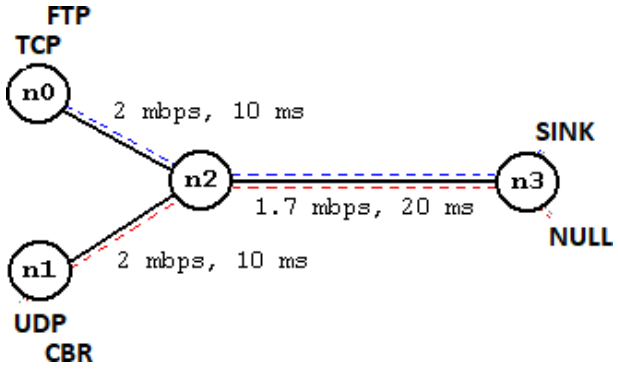


Topology

EXPERIEMENT 4

Simulate a three nodes Point-To-Point Network and Connect the Duplex Links between them. Set the Queue Size, Vary the Transmission Speeds (Bandwidth) and Find the number of Packets Dropped.

Note: Point to point network with the links connected as follows: n0 – n2, n1 – n2 and n2 – n3. Apply TCP agent between n0 – n3 and UDP agent between n1 – n3. Apply relevant applications over TCP and UDP agents changing the parameter and determine the number of packets sent by TCP / UDP.

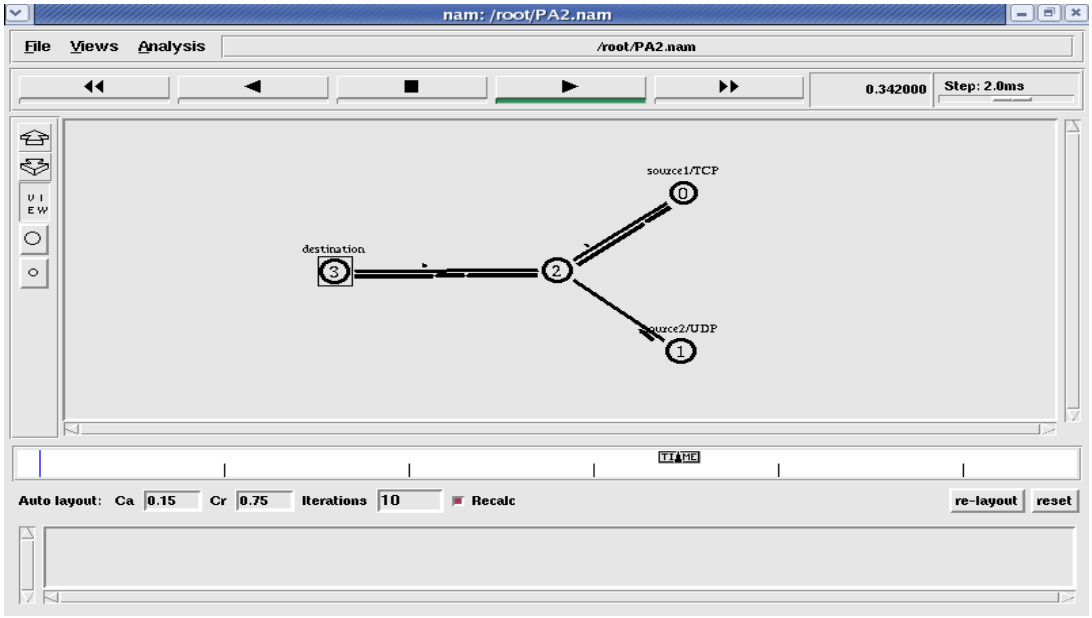


Program:

```
set ns [new Simulator]

set tracefile [open lab2.tr w]
$ns trace-all $tracefile

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

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

set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]

$ns duplex-link $n0  $n2 10.0Mb 1ms DropTail
$ns queue-limit $n0  $n2 10
$ns duplex-link $n1  $n2 10.0Mb 1ms DropTail
$ns queue-limit $n1  $n2 10
$ns duplex-link $n2  $n3 10.0Mb 1ms DropTail
$ns queue-limit $n2  $n3 10

set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
set sink3 [new Agent/TCPSink]
$ns attach-agent $n3 $sink3
$ns connect $tcp0 $sink3
```

```
set udp1 [new Agent/UDP]
$ns attach-agent $n1 $udp1
set null3 [new Agent/Null]
$ns attach-agent $n3 $null3
$ns connect $udp1 $null3


set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
$ns at 1.0 "$ftp0 start"
$ns at 10.0 "$ftp0 stop"

set cbr1 [new Application/Traffic/CBR]
$cbr1 attach-agent $udp1
$ns at 1.0 "$cbr1 start"
$ns at 10.0 "$cbr1 stop"
$ns at 11.0 "finish"
$ns run
```

SIMULATION RESULTS



Topology

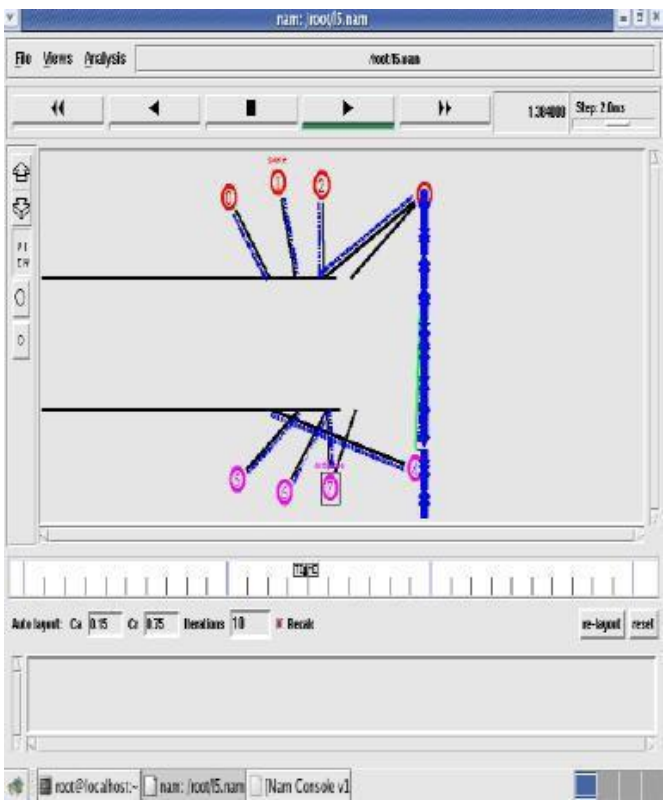
EXPERIEMENT 5

Simulate an Ethernet LAN using n-nodes (6-10) with UDP/TCP connection. Apply relevant applications over UDP/TCP agents by changing the parameters and hence determine the number of packets transmitted.

```
set ns [new Simulator]

set tracefile [open 4.tr w]
$ns trace-all $tracefile

set namfile [open 4.nam w]
$ns namtrace-all $namfile

proc finish {} {
global ns tracefile namfile
$ns flush-trace
close $tracefile
close $namfile
exec nam 4.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]

$ns duplex-link $n0 $n2 10.0Mb 1ms DropTail
$ns queue-limit $n0 $n2 10
$ns duplex-link $n1 $n2 10.0Mb 1ms DropTail
$ns queue-limit $n1 $n2 10
$ns simplex-link $n2 $n3 10.0Mb 1ms DropTail
$ns queue-limit $n2 $n3 10
$ns simplex-link $n3 $n2 10.0Mb 1ms DropTail
$ns queue-limit $n3 $n2 10


$ns duplex-link-op $n0 $n2 orient right-down
$ns duplex-link-op $n1 $n2 orient right-up
$ns simplex-link-op $n2 $n3 orient right
$ns simplex-link-op $n3 $n2 orient left

set lan [$ns newLan "$n3 $n4 $n5" 0.5Mb 40ms LL Queue/DropTailMAC/802_3
   Channel]
set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
set sink4 [new Agent/TCPSink]
$ns attach-agent $n4 $sink4
$ns connect $tcp0 $sink4
$tcp0 set packetSize_ 1500
set udp1 [new Agent/UDP]
$ns attach-agent $n1 $udp1
set null5 [new Agent/Null]
$ns attach-agent $n5 $null5
```

```
$ns connect $udp1 $null5
$udp1 set packetSize_ 1500

set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
$ns at 1.0 "$ftp0 start"
$ns at 10.0 "$ftp0 stop"

set cbr1 [new Application/Traffic/CBR]
$cbr1 attach-agent $udp1

$cbr1 set rate_ 5.0Mb
$ns at 1.0 "$cbr1 start"
$ns at 10.0 "$cbr1 stop"
set myvar [new ErrorModel]

$myvar set rate_ 0.05
$myvar ranvar [new RandomVariable/Uniform]
$myvar drop-target [new Agent/Null]
$ns lossmodel $myvar $n2 $n3

$ns at 11.0 "finish"

$ns run
```

Simulation Results



Topology **Trace file**

Experiment 6.

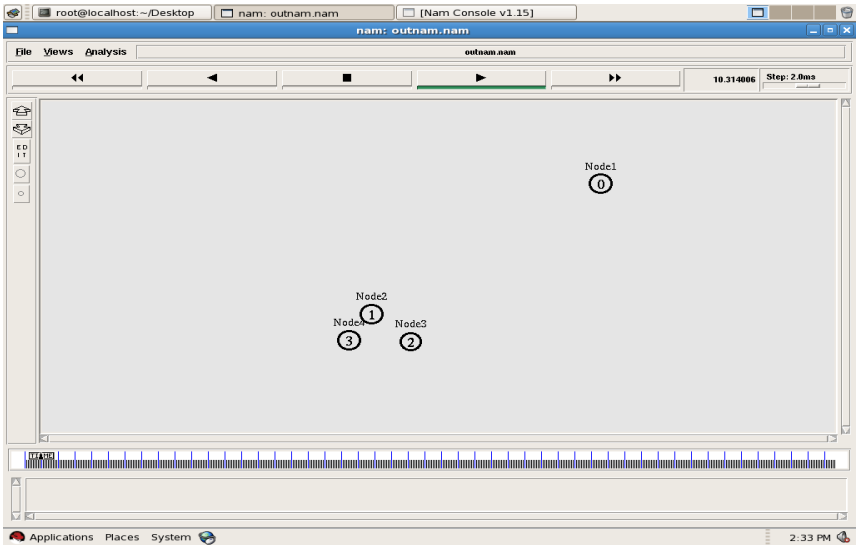
Simulate a wireless network for n nodes. For a wireless network consisting of three mobile nodes(n0-n2), Nodes are configured with the specific parameters of a wireless node. Initial location ofthe node is fixed. Nodes are given mobility with fixed speed and fixed destination location. TCPagent is attached to node0 and TCP sink agent is attached to node1. Both the agents are connected and FTP application is attached to TCP agent. Write a Tcl script and make an ad-hoc simulation to analyze the output in the trace file. Use the routing protocol as Adhoc on demanddistance vector (AODV).

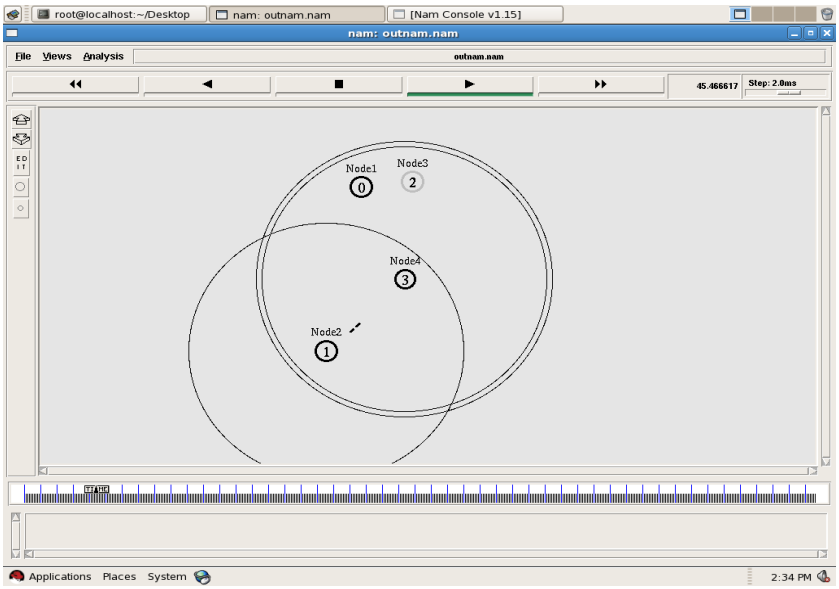
```
#create a new simulator
set ns [new Simulator]

#define options
set val(chan)    Channel/WirelessChannel      ;#channeltype
set val(prop)    Propagation/TwoRayGround     ;#propagation type
set val(ant)     Antenna/OmniAntenna          ;#Antenna type
set val(ll)      LL                      ;#linklayer type
set val(ifq)     Queue/DropTail/PriQueue      ;#interface queue type
set val(ifqlen) 50                       ;#max packet in ifq
set val(netif)   Phy/WirelessPhy              ;#network interface type
set val(mac)     Mac/802_11              ;#MAC type
set val(rp)      AODV                    ;#ad-hoc routing protocol
set val(nn)      10                      ;#number of mobile nodes
set val(x)  500                    ;#x-coordinates of simulation area
set val(y)  500                    ;#y-coordinates of simulation area
set val(stop)    10.0                    ;#time of simulation end

#create topology
set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)

#create trace file
set tf [open outtrace.tr w]
$ns trace-all $tf

#create nam file
set nf [open outnam.nam w]
$ns namtrace-all-wireless $nf $val(x) $val(y)

#create global file
create-god $val(nn)

#configuring the node
$ns node-config  -adhocRouting $val(rp) \
                 -llType $val(ll) \
                 -macType $val(mac) \
                 -ifqType $val(ifq) \
                 -ifqLen $val(ifqlen) \
                 -antType $val(ant) \
                 -propType $val(prop) \
                 -phyType $val(netif) \
                 -channelType $val(chan) \
                 -topoInstance $topo \
                 -agentTrace ON \
                 -routerTrace ON \
                 -macTrace ON \
```

```
                    -movementTrace ON

set node1 [$ns node]
set node2 [$ns node]
set node3 [$ns node]
set node4 [$ns node]

#giving initial node position
$node1 set X_ [expr rand()*$val(x)]
$node1 set Y_ [expr rand()*$val(y)]
$node1 set Z_ 0

$node2 set X_ [expr rand()*$val(x)]
$node2 set Y_ [expr rand()*$val(y)]
$node2 set Z_ 0

$node3 set X_ [expr rand()*$val(x)]
$node3 set Y_ [expr rand()*$val(y)]
$node3 set Z_ 0

$node4 set X_ [expr rand()*$val(x)]
$node4 set Y_ [expr rand()*$val(y)]
$node4 set Z_ 0

$ns at 0.1 "$node1 label Node1"
$ns at 0.1 "$node2 label Node2"
$ns at 0.1 "$node3 label Node3"
$ns at 0.1 "$node4 label Node4"

#giving node size
$ns initial_node_pos $node1 30
$ns initial_node_pos $node2 30
$ns initial_node_pos $node3 30
$ns initial_node_pos $node4 30

set tcp1 [new Agent/TCP]
$tcp1 set class_ 2
$ns attach-agent $node1 $tcp1
set sink1 [new Agent/TCPSink]
$ns attach-agent $node2 $sink1
$ns connect $tcp1 $sink1
set ftp [new Application/FTP]
$ftp attach-agent $tcp1

#giving mobility to nodes
$ns at 10.0 "$node1 setdest [expr rand()*$val(x)] [expr rand()*$val(y)] 30.0"
$ns at 10.0 "$node2 setdest [expr rand()*$val(x)] [expr rand()*$val(y)] 30.0"
$ns at 10.0 "$node4 setdest [expr rand()*$val(x)] [expr rand()*$val(y)] 15.0"
$ns at 20.0 "$node4 setdest [expr rand()*$val(x)] [expr rand()*$val(y)] 15.0"
$ns at 30.0 "$node3 setdest [expr rand()*$val(x)] [expr rand()*$val(y)] 15.0"
$ns at 40.0 "$node4 setdest [expr rand()*$val(x)] [expr rand()*$val(y)] 15.0"
$ns at 50.0 "$node3 setdest [expr rand()*$val(x)] [expr rand()*$val(y)] 15.0"
$ns at 60.0 "$node4 setdest [expr rand()*$val(x)] [expr rand()*$val(y)] 15.0"
$ns at 70.0 "$node3 setdest [expr rand()*$val(x)] [expr rand()*$val(y)] 15.0"

$ns at 0.5 "$ftp start"
$ns at 500 "$ftp stop"

$ns at 500 "finish"
```

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

Output :





MODEL VIVA QUESTIONS

1) What is a Link?

A link refers to the connectivity between two devices. It includes the type of cables and protocols used in order for one device to be able to communicate with the other.

2) What are the layers of the OSI reference model?

There are 7 OSI layers: Physical Layer, Data Link Layer, Network Layer, Transport Layer, Session Layer, Presentation Layer and Application Layer.

3) What is backbone network?

A backbone network is a centralized infrastructure that is designed to distribute different routes and data to various networks. It also handles management of bandwidth and various channels.

4) What is a LAN?

LAN is short for Local Area Network. It refers to the connection between computers and other network devices that are located within a small physical location.

5) What is a node?

A node refers to a point or joint where a connection takes place. It can be computer or device that is part of a network. Two or more nodes are needed in order to form a network connection.

6) What are routers?

Routers can connect two or more network segments. These are intelligent network devices that store information in its routing table such as paths, hops and bottlenecks. With this info, they are able to determine the best path for data transfer. Routers operate at the OSI Network Layer.

7) What is point to point link?

It refers to a direct connection between two computers on a network. A point to point connection does not need any other network devices other than connecting a cable to the NIC cards of both computers.

8) What is anonymous FTP?

Anonymous FTP is a way of granting user access to files in public servers. Users that are allowed access to data in these servers do not need to identify themselves, but instead log in as an anonymous guest.

9) What is subnet mask?

A subnet mask is combined with an IP address in order to identify two parts: the extended network address and the host address. Like an IP address, a subnet mask is made up of 32 bits.

10) What is the maximum length allowed for a UTP cable?

A single segment of UTP cable has an allowable length of 90 to 100 meters. This limitation can be overcome by using repeaters and switches.

11) What is data encapsulation?

Data encapsulation is the process of breaking down information into smaller manageable chunks before it is transmitted across the network. It is also in this process that the source and destination addresses are attached into the headers, along with parity checks.

12) Describe Network Topology

Network Topology refers to the layout of a computer network. It shows how devices and cables are physically laid out, as well as how they connect to one another.

13) What is VPN?

VPN means Virtual Private Network, a technology that allows a secure tunnel to be created across a network such as the Internet. For example, VPNs allow you to establish a secure dialup connection to a remote server.

14) Briefly describe NAT.

NAT is Network Address Translation. This is a protocol that provides a way for multiple computers on a common network to share single connection to the Internet.

15) What is the job of the Network Layer under the OSI reference model?

The Network layer is responsible for data routing, packet switching and control of network congestion. Routers operate under this layer.

16) How does a network topology affect your decision in setting up a network?

Network topology dictates what media you must use to interconnect devices. It also serves as basis on what materials, connector and terminations that is applicable for the setup.

17) What is RIP?

RIP, short for Routing Information Protocol is used by routers to send data from one network to another. It efficiently manages routing data by broadcasting its routing table to all other routers within the network. It determines the network distance in units of hops.

18) What are different ways of securing a computer network?

There are several ways to do this. Install reliable and updated anti-virus program on all computers. Make sure firewalls are setup and configured properly. User authentication will also help a lot. All of these combined would make a highly secured network.

19) What is NIC?

NIC is short for Network Interface Card. This is a peripheral card that is attached to a PC in order to connect to a network. Every NIC has its own MAC address that identifies the PC on the network.

20) What is WAN?

WAN stands for Wide Area Network. It is an interconnection of computers and devices that are geographically dispersed. It connects networks that are located in different regions and countries.

21) What is the importance of the OSI Physical Layer?

The physical layer does the conversion from data bits to electrical signal, and vice versa. This is where network devices and cable types are considered and setup.

22) How many layers are there under TCP/IP?

There are four layers: the Network Layer, Internet Layer, Transport Layer and Application Layer.

23) What are proxy servers and how do they protect computer networks

Proxy servers primarily prevent external users who identifying the IP addresses of an internal network. Without knowledge of the correct IP address, even the physical location of the network cannot be identified. Proxy servers can make a network virtually invisible to external users.

24) What is the function of the OSI Session Layer?

This layer provides the protocols and means for two devices on the network to communicate with each other by holding a session. This includes setting up the session, managing information exchange during the session, and tear-down process upon termination of the session.

25) What is the importance of implementing a Fault Tolerance System? Are there limitations?

A fault tolerance system ensures continuous data availability. This is done by eliminating a single point of failure. However, this type of system would not be able to protect data in some cases, such as in accidental deletions.

26) What does 10Base-T mean?

The 10 refers to the data transfer rate, in this case is 10Mbps. The word Base refers to base band, as oppose to broad band. T means twisted pair, which is the cable used for that network.

27) What is a private IP address?

Private IP addresses are assigned for use on intranets. These addresses are used for internal networks and are not routable on external public networks. These ensures that no conflicts are present among internal networks while at the same time the same range of private IP addresses are reusable for multiple intranets since they do not "see" each other.

28) What is NOS?

NOS, or Network Operating System, is specialized software whose main task is to provide network connectivity to a computer in order for it to be able to communicate with other computers and connected devices.

29) What is Do's?

Do's, or Denial-of-Service attack, is an attempt to prevent users from being able to access the internet or any other network services. Such attacks may come in different forms and are done by a group of perpetuators. One common method of doing this is to overload the system server so it cannot anymore process legitimate traffic

And will be forced to reset.

30) What is OSI and what role does it play in computer networks?

OSI (Open Systems Interconnect) serves as a reference model for data communication. It is made up of 7 layers, with each layer defining a particular aspect on how network devices connect and communicate with one another. One layer may deal with the physical media used, while another layer dictates how data is actually transmitted across the network.

31) What is the purpose of cables being shielded and having twisted pairs?

The main purpose of this is to prevent crosstalk. Crosstalks are electromagnetic interferences or noise that can affect data being transmitted across cables.

32) What is the advantage of address sharing?

By using address translation instead of routing, address sharing provides an inherent security benefit. That's because host PCs on the Internet can only see the public IP address of the external interface on the computer that provides address translation and not the private IP addresses on the internal network.

33) What are MAC addresses?

MAC, or Media Access Control, uniquely identifies a device on the network. It is also known as physical address or Ethernet address. A MAC address is made up of 6-byte parts.

34) What is the equivalent layer or layers of the TCP/IP Application layer in terms of OSI reference model?

The TCP/IP Application layer actually has three counterparts on the OSI model: the Session layer, Presentation Layer and Application Layer.

35) How can you identify the IP class of a given IP address?

By looking at the first octet of any given IP address, you can identify whether it's Class A, B or C. If the first octet begins with a 0 bit, that address is Class A. If it begins with bits 10 then that address is a Class B address. If it begins with 110, then it's a Class C network.

36) What is the main purpose of OSPF?

OSPF, or Open Shortest Path First, is a link-state routing protocol that uses routing tables to determine the best possible path for data exchange.

37) What are firewalls?

Firewalls serve to protect an internal network from external attacks. These external threats can be hackers who want to steal data or computer viruses that can wipe out data in an instant. It also prevents other users from external networks from gaining access to the private network.

38) Describe star topology

Star topology consists of a central hub that connects to nodes. This is one of the easiest to setup and maintain.

39) What are gateways?

Gateways provide connectivity between two or more network segments. It is usually a computer that runs the gateway software and provides translation services. This translation is a key in allowing different systems to communicate on the network.

40) What is the disadvantage of a star topology?

One major disadvantage of star topology is that once the central hub or switch get damaged, the entire network becomes unusable.

41) What is SLIP?

SLIP, or Serial Line Interface Protocol, is actually an old protocol developed during the early UNIX days. This is one of the protocols that are used for remote access.

42) Give some examples of private network addresses.

10.0.0.0 with a subnet mask of 255.0.0.0

172.16.0.0 With subnet mask of 255.240.0.0

192.168.0.0 with subnet mask of 255.255.0.0

43) What is tracer?

Tracer is a Windows utility program that can used to trace the route taken by data from the router to the destination network. It also shows the number of hops taken during the entire transmission route.

44) What are the functions of a network administrator?

A network administrator has many responsibilities that can be summarize into 3 key functions: installation of a network, configuration of network settings, and maintenance/troubleshooting of networks.

45) Describe at one disadvantage of a peer to peer network.

When you are accessing the resources that are shared by one of the workstations on the network, that workstation takes a performance hit.

46) What is Hybrid Network?

A hybrid network is a network setup that makes use of both client-server and peer-to-peer architecture.

47) What is DHCP?

DHCP is short for Dynamic Host Configuration Protocol. Its main task is to automatically assign an IP address to devices across the network. It first checks for the next available address not yet taken by any device, then assigns this to a network device.

48) What is the main job of the ARP?

The main task of ARP or Address Resolution Protocol is to map a known IP address to a MAC layer address.

49) What is TCP/IP?

TCP/IP is short for Transmission Control Protocol / Internet Protocol. This is a set of protocol layers that is designed to make data exchange possible on different types of computer networks, also known as heterogeneous network.

50) How can you manage a network using a router?

Routers have built in console that lets you configure different settings, like security and data logging. You can assign restrictions to computers, such as what resources it is allowed access, or what particular time of the day they can browse the internet. You can even put restrictions on what websites are not viewable across the entire network.

51) What protocol can be applied when you want to transfer files between different platforms, such between UNIX systems and Windows servers?

Use FTP (File Transfer Protocol) for file transfers between such different servers. This is possible because FTP is platform independent.

52) What is the use of a default gateway?

Default gateways provide means for the local networks to connect to the external network. The default gateway for connecting to the external network is usually the address of the external router port.

53) One way of securing a network is through the use of passwords. What can be considered as good passwords?

Good passwords are made up of not just letters, but by combining letters and numbers. A password that combines uppercase and lowercase letters is favorable than one that uses all upper case or all lower case letters. Passwords must be not words that can easily be guessed by hackers, such as dates, names, favorites, etc. Longer passwords are also better than short ones.

54) What is the proper termination rate for UTP cables?

The proper termination for unshielded twisted pair network cable is 100 ohms.

55) What is net stat?

Net stat is a command line utility program. It provides useful information about the current TCP/IP settings of a connection.

56) What is the number of network IDs in a Class C network?

For a Class C network, the number of usable Network ID bits is 21. The number of possible network IDs is 2 raised to 21 or 2,097,152. The number of host IDs per network ID is 2 raised to 8 minus 2, or 254.

57) What happens when you use cables longer than the prescribed length?

Cables that are too long would result in signal loss. This means that data transmission and reception would be affected, because the signal degrades over length.

58) What common software problems can lead to network defects?

Software related problems can be any or a combination of the following:

- client server problems

- application conflicts

- error in configuration

- protocol mismatch

- security issues

- user policy and rights issues

59) What is ICMP?

ICMP is Internet Control Message Protocol. It provides messaging and communication for protocols within the TCP/IP stack. This is also the protocol that manages error messages that are used by network tools such as PING.

60) What is Ping?

Ping is a utility program that allows you to check connectivity between network devices on the network. You can ping a device by using its IP address or device name, such as a computer name.

61) What is peer to peer?

Peer to peer are networks that does not reply on a server. All PCs on this network act as individual workstations.

62) What is DNS?

DNS is Domain Name System. The main function of this network service is to provide host names to TCP/IP address resolution.

63) What advantages does fiber optics have over other media?

One major advantage of fiber optics is that is it less susceptible to electrical interference. It also supports higher bandwidth, meaning more data can be transmitted and received. Signal degrading is also very minimal over long distances.

64) What is the difference between a hub and a switch?

A hub acts as a multiport repeater. However, as more and more devices connect to it, it would not be able to efficiently manage the volume of traffic that passes through it. A switch provides a better alternative that can improve the performance especially when high traffic volume is expected across all ports.

65) What are the different network protocols that are supported by Windows RRAS services?

There are three main network protocols supported: NetBEUI, TCP/IP, and IPX.

66) What are the maximum networks and hosts in a class A, B and C network?

For Class A, there are 126 possible networks and 16,777,214 hosts

For Class B, there are 16,384 possible networks and 65,534 hosts

For Class C, there are 2,097,152 possible networks and 254 hosts

67) What is the standard color sequence of a straight-through cable?

Orange/white, orange, green/white, blue, blue/white, green, brown/white, brown.

68) What protocols fall under the Application layer of the TCP/IP stack?

The following are the protocols under TCP/IP Application layer: FTP, TFTP, Telnet and SMTP.

69) You need to connect two computers for file sharing. Is it possible to do this without using a hub or router?

Yes, you can connect two computers together using only one cable. A crossover type cable can be used in this scenario. In this setup, the data transmit pin of one cable is connected to the data receive pin of the other cable, and vice versa.

70) What is ipconfig?

Ipconfig is a utility program that is commonly used to identify the addresses information of a computer on a network. It can show the physical address as well as the IP address.

71) What is the difference between a straight-through and crossover cable?

A straight-through cable is used to connect computers to a switch, hub or router. A crossover cable is used to connect two similar devices together, such as a PC to PC or Hub to hub.

72) What is client/server?

Client/server is a type of network wherein one or more computers act as servers. Servers provide a centralized repository of resources such as printers and files. Clients refers to workstation that access the server.

73) Describe networking.

Networking refers to the inter connection between computers and peripherals for data communication.

Networking can be done using wired cabling or through wireless link.

74) When you move the NIC cards from one PC to another PC, does the MAC address gets transferred as well?

Yes, that's because MAC addresses are hard-wired into the NIC circuitry, not the PC. This also means that a PC can have a different MAC address when the NIC card was replace by another one.

75) Explain clustering support

Clustering support refers to the ability of a network operating system to connect multiple servers in a fault-tolerant group. The main purpose of this is the in the event that one server fails, all processing will continue on with the next server in the cluster.

76) In a network that contains two servers and twenty workstations, where is the best place to install an Anti-virus program?

An anti-virus program must be installed on all servers and workstations to ensure protection. That's because individual users can access any workstation and introduce a computer virus when plugging in their removable hard drives or flash drives.

77) Describe Ethernet.

Ethernet is one of the popular networking technologies used these days. It was developed during the early 1970s and is based on specifications as stated in the IEEE. Ethernet is used in local area networks.

78) What are some drawbacks of implementing a ring topology?

In case one workstation on the network suffers a malfunction, it can bring down the entire network. Another drawback is that when there are adjustments and reconfigurations needed to be performed on a particular part of the network, the entire network has to be temporarily brought down as well.

79) What is the difference between CSMA/CD and CSMA/CA?

CSMA/CD, or Collision Detect, retransmits data frames whenever a collision occurred. CSMA/CA, or Collision Avoidance, will first broadcast intent to send prior to data transmission.

80) What is SMTP?

SMTP is short for Simple Mail Transfer Protocol. This protocol deals with all Internal mail, and provides the necessary mail delivery services on the TCP/IP protocol stack.

81) What is multicast routing?

Multicast routing is a targeted form of broadcasting that sends message to a selected group of user, instead of sending it to all users on a subnet.

82) What is the importance of Encryption on a network?

Encryption is the process of translating information into a code that is unreadable by the user. It is then translated back or decrypted back to its normal readable format using a secret key or password. Encryption help ensure that information that is intercepted halfway would remain unreadable because the user has to have the correct password or key for it.

83) How are IP addresses arranged and displayed?

IP addresses are displayed as a series of four decimal numbers that are separated by period or dots. Another term for this arrangement is the dotted decimal format. An example is 192.168.101.2

84) Explain the importance of authentication.

Authentication is the process of verifying a user's credentials before he can log into the network. It is normally performed using a username and password. This provides a secure means of limiting the access from unwanted intruders on the network.

85) What do mean by tunnel mode?

This is a mode of data exchange wherein two communicating computers do not use IPsec themselves. Instead, the gateway that is connecting their LANs to the transit network creates a virtual tunnel that uses the IPsec protocol to secure all communication that passes through it.

86) What are the different technologies involved in establishing WAN links?

Analog connections - using conventional telephone lines; Digital connections - using digital grade telephone lines; switched connections - using multiple sets of links between sender and receiver to move data.

87) What is one advantage of mesh topology?

In the event that one link fails, there will always be another available. Mesh topology is actually one of the most

Fault-tolerant network topology.

88) When troubleshooting computer network problems, what common hardware-related

problems can occur?

A large percentage of a network is made up of hardware. Problems in these areas can range from malfunctioning hard drives, broken NICs and even hardware startups. Incorrectly hardware configuration is also one of those culprits to look into.

89) What can be done to fix signal attenuation problems?

A common way of dealing with such a problem is to use repeaters and hub, because it will help regenerate the signal and therefore prevent signal loss. Checking if cables are properly terminated is also a must.

90) How does dynamic host configuration protocol aid in network administration?

Instead of having to visit each client computer to configure a static IP address, the network administrator can apply dynamic host configuration protocol to create a pool of IP addresses known as scopes that can be dynamically assigned to clients.

91) Explain profile in terms of networking concept?

Profiles are the configuration settings made for each user. A profile may be created that puts a user in a group, for example.

92) What is sneaker net?

Sneaker net is believed to be the earliest form of networking wherein data is physically transported using removable media, such as disk, tapes.

93) What is the role of IEEE in computer networking?

IEEE, or the Institute of Electrical and Electronics Engineers, is an organization composed of engineers that issues and manages standards for electrical and electronic devices. This includes networking devices, network interfaces, cablings and connectors.

94) What protocols fall under the TCP/IP Internet Layer?

There are 4 protocols that are being managed by this layer. These are ICMP, IGMP, IP and ARP.

95) When it comes to networking, what are rights?

Rights refer to the authorized permission to perform specific actions on the network. Each user on the network can be assigned individual rights, depending on what must be allowed for that user.

96) What is one basic requirement for establishing VLANs?

A VLAN requires dedicated equipment on each end of the connection that allows messages entering the Internet to be encrypted, as well as for authenticating users.

97) What is IPv6?

IPv6, or Internet Protocol version 6, was developed to replace IPv4. At present, IPv4 is being used to control internet traffic, butis expected to get saturated in the near future. IPv6 was designed to overcome this limitation.

98) What is RSA algorithm?

RSA is short for Rivets-Shamir-Adelman algorithm. It is the most commonly used public key encryption algorithm in use today.

99) What is mesh topology?

Mesh topology is a setup wherein each device is connected directly to every other device on the network. Consequently, it requires that each device have at least two network connections.