

**EC0421 NETWORK SIMULATION LAB**

**LABORATORY MANUAL**

**SEMESTER VII**



**DEAPRTMENT OF  
ELECTRONICS AND COMMUNICATION ENGINEERING  
SRM UNIVERISTY  
(Under SECTION 3 of the UGC Act, 1956)  
S.R.M. NAGAR, KATTANKULATHUR – 603203.  
KANCHEEPURAM DISTRICT**



**Department of Electronics and Communication Engineering**

**EC0421**  
**Network Simulation Lab**

**Laboratory Manual**

**Course Team**

**Dr. V. Nithya**  
**Mrs. R. Vinolee**  
**Mrs. R. Dayana**  
**Mr. E. Elamaran**  
**Dr. M. Sangeetha**  
**Mrs. Sabitha Gauni**  
**Mr. S. Praveenkumar**  
**Mr . S. Manikandansway**

**June 2015**

**Revision: 2**

<b>EC0421</b>	<b>Network Simulation Lab</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>0</b>	<b>0</b>	<b>3</b>	<b>2</b>

### PURPOSE

To know and understand communication networks using NETSIM Software and LAN Trainer kit.

### INSTRUCTIONAL OBJECTIVES

To study the communication networks characteristics and to analyze various MAC and routing layer Protocols.

### LIST OF EXPERIMENTS

1. Ethernet LAN protocol. To create Scenario and study the performance of CSMA/CD protocol through simulation
2. Token Bus and Token Ring protocols. To create scenario and study the performance of token bus and token ring protocols through simulation
3. Wireless LAN protocols. To create scenario and study the performance of network with CSMA/CA protocol and compare with CSMA/CD protocols
4. Implementation and study of Stop and Wait protocol
5. Implementation and study of Go back N and Selective Repeat protocols
6. Implementation of Distance Vector Routing algorithm
7. Implementation of Link state routing algorithm
8. Implementation of data encryption and decryption
9. Transfer of files from PC to PC using Windows/ UNIX socket processing

**TOTAL 45**

### REFERENCE: LABORATORY MANUAL

#### EC0421-Network Simulation Lab

Course designed by	Department of Electronics & Communication Engineering										
Program outcome	a	b	c	d	e	f	g	h	i	j	k
			X	X		X		X		X	X
Category	General (G)	Basic Sciences (B)	Engineering Sciences and Technical Arts(E)				Professional Subjects(P)				
							X				
Broad area (for 'P' category)	Communication	Signal Processing	Electronics				VLSI	Embedded			
	X										
Staff responsible for preparing the syllabus	Ms.V.Nithya										
Date of preparation	December 2006										

**S.R.M University**  
**Faculty of Engineering and Technology**  
**Department of Electronics and Communication Engineering**

**Sub Code: EC0421**  
**Sub Title: Network Simulation Lab**  
**Pre\_requisite : Nil**  
**Co\_requisite : Nil**

**Semester : VII**  
**Course Time: Jun -Dec'15**

**Program Outcome**

**c. an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability**

Experiment: 1-5: The practical throughput and the offered load for different LAN Topologies are calculated by varying the parameters like BER, Inter packet delay etc .The obtained data are analyzed and interpreted using plots.

**d. an ability to function on multidisciplinary teams**

Experiment: 1-9: The experiments are performed by configuring the nodes to identify the network parameters such as the number of collisions, successfully transmitted packets, link failure, CRC errors etc., and the throughput is calculated using the formula to analyze and solve the network engineering problems

**f. an understanding of professional and ethical responsibility**

Experiment: 1-9: The experiments are performed by configuring the nodes to identify the network parameters such as the number of collisions, successfully transmitted packets, link failure, CRC errors etc., and the throughput is calculated using the formula to analyze and solve the network engineering problems

**h. the broad education necessary to understand the impact of engineering solutions in a global, economic, environmental, and societal context**

Experiment: 1-5: The practical throughput and the offered load for different LAN Topologies are calculated by varying the parameters like BER, Inter packet delay etc .The obtained data are analyzed and interpreted using plots.

**(j) a knowledge of contemporary issues**

Experiment: 9: enables reliable transfer of files from one PC to another using Windows socket processing.

**(k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.**

Experiment: 6, 7, 8 and 9: helps in using the NetSim software to analyze the various routing algorithms

All the experiments carried out in the lab aim in knowing the basic principles relating to network and solving network problems.

**Sub Code: EC0421**  
**Sub Title: Network Simulation Lab**

**Semester : VII**  
**Course Time: Jun-Dec'15**

**Pre\_requisite : Nil**  
**Co\_requisite : Nil**

### **Program Educational Objectives vs Program Outcome**

<b>Program Educational Objective</b> → <b>Student Outcomes</b> ↓	<b>PEO1:</b> Graduates will perform as a successful professional engineer in related fields of Electronics and Communication Engineering.	<b>PEO2:</b> Graduates will pursue higher education and/or engage themselves in continuous professional development to meet global standards.	<b>PEO3:</b> Graduates will work as a team in diverse fields and gradually move into leadership positions.	<b>PEO4:</b> Graduates will understand current professional issues, apply latest technologies and come out with innovative solutions for the betterment of the nation and society.
<b>c</b>	√			√
<b>d</b>	√		√	√
<b>f</b>	√	√	√	√
<b>h</b>	√	√		√
<b>j</b>	√	√		√
<b>k</b>	√	√		√

Sub Code: EC0421  
 Sub Title: Network Simulation Lab  
 Pre\_requisite : Nil  
 Co\_requisite : Nil

Semester : VII  
 Course Time: Jun–Dec'14

### Instructional Objective and Program Outcome

S.No.	Instructional Objective	Program Outcome	Experiment Details
1	To study the communication networks characteristics and to analyze various MAC and routing layer Protocols.	<p>(c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability</p> <p>(d) an ability to function on multidisciplinary teams</p> <p>(f) an understanding of professional and ethical responsibility</p> <p>(h) the broad education necessary to understand the impact of engineering solutions in a global, economic, environmental, and societal context</p> <p>(j) a knowledge of contemporary issues</p> <p>(k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.</p>	<ol style="list-style-type: none"> <li>1. Ethernet LAN protocol: To create a network scenario and study the performance of CSMA/CD protocol through simulation</li> <li>2. Token Bus and Token Ring protocols: To create a network scenario and study the performance of token bus and token ring protocols through simulation</li> <li>3. Wireless LAN protocols: To create a network scenario and study the performance of network with CSMA/CA protocol and compare with CSMA/CD protocols</li> <li>4. Implementation and study of Stop and Wait protocol</li> <li>5. Implementation and study of Go back N and Selective Repeat protocols</li> <li>6. Implementation of Distance Vector Routing algorithm</li> <li>7. Implementation of Link state routing algorithm</li> <li>8. Implementation of data encryption and decryption</li> <li>9. Transfer of files from PC to PC using Windows/UNIX socket processing</li> </ol>

**S.R.M University**  
**Faculty of Engineering and Technology**  
**Department of Electronics and Communication Engineering**

**Sub Code: EC0421**  
**Sub Title: Network Simulation Lab**

**Semester : VII**  
**Course Time : Jul–Dec'15**

**Pre\_requisite : Nil**  
**Co\_requisite : Nil**

**EXPERIMENTS DETAILS**

<b>S.No.</b>	<b>Experiments Detail</b>	<b>Equipments Required</b>
1	Ethernet LAN protocol. To create Scenario and study the performance of CSMA/CD protocol through simulation.	LAN Trainer Kit and three Computers
2	Token Bus and Token Ring protocols. To create scenario and study the performance of token bus and token ring protocols through simulation	LAN Trainer Kit and three Computers
3	Wireless LAN protocols. To create scenario and study the performance of network with CSMA/CA protocol and compare with CSMA/CD protocols	LAN Trainer Kit and three Computers
4	Implementation and study of Stop and Wait protocol	LAN Trainer Kit and three Computers
5	Implementation and study of Go back N and Selective Repeat protocols	LAN Trainer Kit and Computers
6	Implementation of Distance Vector Routing algorithm	Computers with Net Sim Software
7	Implementation of Link state routing algorithm	Computers with Net Sim Software
8	Implementation of data encryption and decryption	Two Computers with Net Sim software
9	Transfer of files from PC to PC using Windows/ UNIX socket processing	Two Computers with Net Sim software

# Specification of LAN Trainer Kit

## Experiment Software

- LAN Trainer Shell – provides a menu driven interface to the experiments.
- ‘C’ library - Programming interface to the NIU.
- Stand alone programs for observation experiments. Source code provided for study and analysis.
- Saving of experiment results to a file.
- Display of experiment activities in the application window, wherever applicable.

## Hardware

### Network Emulator Unit (NEU)

- Data rates: 8, 16, 32, 64, 128, 256, 512Kbps, 1Mbps.
- Topology: Bus, Ring, Star.
- Delay: 0 . . . . 15 bits between each pair of nodes.
- Error Generators: Bit error – 0 to  $10^{-6}$   
(between one pair of nodes) Frame error – 0 to  $10^{-5}$
- Nodes: 6 nodes per NEU (3 PCs can be connected per NEU.
- Each PC acts as 2 nodes)
- NEUs can be cascaded to emulate larger networks.

### Network Interface Unit (NIU)

- PC plug in card: ISA bus.
- MAC Layer support: ALOHA, CSMA, CSMA/CD, Token bus, Token Ring.
- Nodes: 2 nodes per NIU.

### System Requirements

- PC: Pentium or higher
- One ISA slot required
- 16MB RAM
- Operating System: Windows 95
- Number of PCs: 2 – 3
- Visual C++ compiler: Version 5.0 or above
- (Optional – for programming/modifying experiments at Level 2 and for further development)



## EC0421 Laboratory Policies and Report Format

Reports are due at the beginning of the lab period. The reports are intended to be a complete documentation of the work done in preparation for and during the lab. The report should be complete so that someone else familiar with computer networks could use it to verify your work. The pre lab and post lab report format is as follows:

1. A neat thorough pre lab must be presented to your Staff In charge at the beginning of your scheduled lab period. **Lab reports should be submitted on A4 paper.** Your report is a professional presentation of your work in the lab. Neatness, organization, and completeness will be rewarded. Points will be deducted for any part that is not clear.

2. In this laboratory students will work in teams of four. However, the lab reports will be written individually. Please use the following format for your lab reports.

a. **Cover Page:** Include your name, Subject Code, Section No., Experiment No. and Date.

b. **Objectives:** Enumerate 3 or 4 of the topics that you think the lab will teach you. There should be one or two sentences per objective. Remember, you should write about what you will learn, not what you will do.

c. **Configuration:** This part contains the configuration details of the node used for the simulation. This should include Configuration menu table where the values of network simulation parameters, type of protocols used are given here. **This section should also include a clear written description of your simulation process.** Simply including a configuration table is not sufficient.

d. **Simulation Results:** The analysis of the given protocol is studied with the help of simulation results. Use the necessary formulas to study the performance of the given protocol and draw necessary plots with the obtained results. The model graph should also be given. Make sure that the neat graph must be drawn for the obtained results.

e. **Questions:** Specific questions (Pre lab and Post lab) asked in the lab should be answered here. **Retype the questions presented in the lab and then formally answer them.**

3. Your work must be original and prepared independently. However, if you need any guidance or have any questions or problems, please do not hesitate to approach your faculty in charge during office hours. Copying any pre lab /post lab will result in a grade of 0. The incident will be formally reported to the University and the students should follow the dress code in the Lab session.

4. Each laboratory experiment must be completed and demonstrated to your faculty in charge in order to receive working module credit. This is the procedure to follow:

a. Expected simulation results obtained: If the experiment is completed with expected simulation results during the lab period (3 hours), call your

Staff in charge, and he/she will sign and date it. This is the end of this lab, and you will get a complete grade for this portion of the lab.

b. No Expected simulation results: If the experiment is done but not with expected results, you must make use of the open times for the lab room to complete your experiment. When your simulation results are correct, contact your staff in charge to set up a time when the two of you can meet to check your results.

5. Attendance at your regularly scheduled lab period is required. An unexpected absence will result in loss of credit for your lab. If for valid reason a student misses a lab, or makes a reasonable request in advance of the class meeting, it is permissible for the student to do the lab in a different section later in the week if approved by the staff in charge of both the sections. Habitually late students (i.e., students late more than 15 minutes more than once) will receive 10 point reductions in their grades for each occurrence following the first.

6. Final grade in this course will be based on laboratory assignments. All labs have an equal weight in the final grade. Grading will be based on pre-lab work, laboratory reports, post-lab and in-lab performance (i.e., completing lab, answering laboratory related questions, etc.). The faculty in charge will ask pertinent questions to individual members of a team at random. Labs will be graded as per the following grading policy:

Pre-Lab Work	10.00%
In-Lab Performance	20.00%
Post Lab Work	10.00%
Laboratory Report	10.00%

8. **Reports Due Dates:** Reports are due one week after completion of the corresponding lab.

9. **Systems of Tests:** Regular laboratory class work over the full semester will carry a weightage of 75%. The remaining 25% weightage will be given by conducting an end semester practical examination for every individual student if possible or by conducting a 1 to 1 ½ hours duration common written test for all students, based on all the experiment carried out in the semester.

#### **10. General Procedure:**

- Properly place the patch cord in its respective position for good internal connection in the trainer kit
- Make sure with the connection of patch cords in the trainer kit in respect to the experiment concerned.
- Do not meddle with the trainer kit
- Run the Diagnostics procedure from one of the PCs connected to the NEU.
- The driver file name and path (C:\Lantrain\Bin\LANTV13.exe) should not be changed.
- Click Reset button to reset NIU

- g. Repeat the above steps from (d to f) to other PCs (one PC at a time) connected to NEU to confirm the LAN-T working.
- h. Set up the bit delay and bit error rate according to the experiment requirement.

**SRM UNIVERSITY**  
**Department of Electronics and Communication Engineering**

**EC0421 Network Simulation Lab**  
**Laboratory Report Cover Sheet**

**ODD SEM – 2015**

Name: \_\_\_\_\_

Section: Tick One M ( )      TU ( )      W ( )      Th ( )      Fr ( )

Venue:                      Networking Lab

Title of Lab:              EC0421 Network Simulation Lab

**Preparation Verification**

Staff Name & Signature: \_\_\_\_\_

\_\_\_\_\_

**Experiment Completion Verification**

Staff Name & Signature: \_\_\_\_\_

\_\_\_\_\_

Date, Time: \_\_\_\_\_

Particulars	Max Marks	Marks Obtained
Pre-lab Work	10	
In Lab Performance	20	
Post lab Work	10	
Lab Report	10	
Total	50	

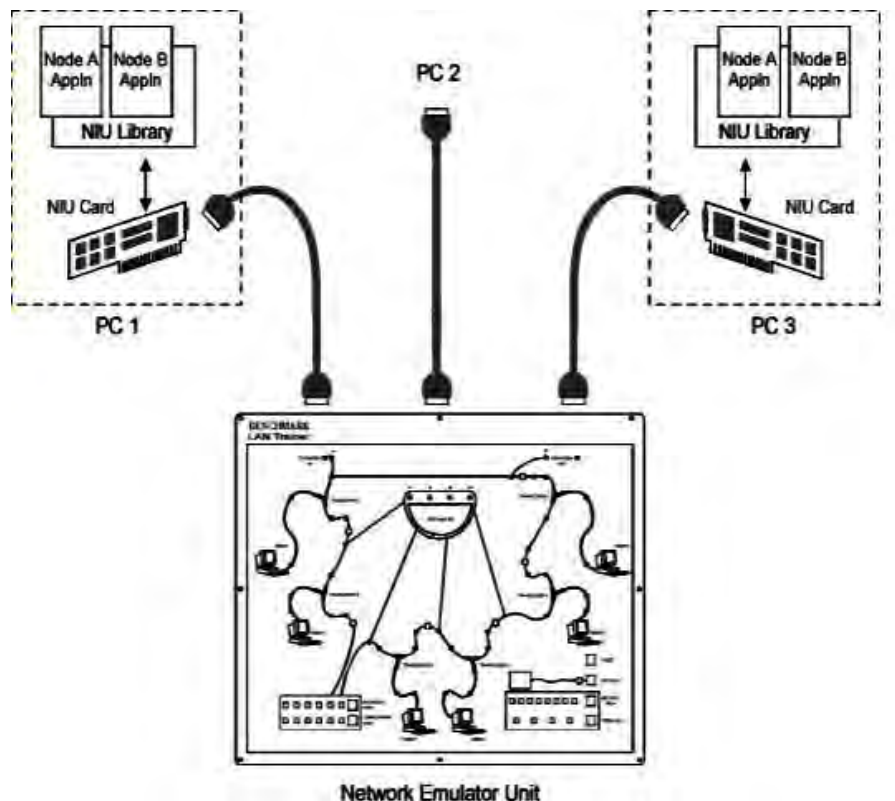
**Report Verification**

Staff Name & Signature: \_\_\_\_\_

\_\_\_\_\_

Date, Time: \_\_\_\_\_

## LAN Trainer KIT Operation



**Figure 1.1:** Components in the LAN Trainer

### **Introduction to the LAN Trainer:**

To successfully use the LAN Trainer, a number of hardware and software components must be properly used together (see Fig. 1.1). Each PC acts as two nodes in the network. Both are connected to the Network Emulator Unit via the same cable. Thus, with 3 PCs, you can experiment with a 6-node network. On the software side, the screen is divided into 2 windows, one for each node. This is accomplished using the LAN Trainer control panel running under Windows – 95

**NIU Card:** Each PC that is part of the LAN Trainer setup must have an NIU Card plugged into it.

### **Network Emulator Unit**

The Network Emulator Unit acts as a network interconnecting up to 6 network nodes (3 NIU Cards in 3 PCs). Each card is connected to the Network Emulator Unit via a cable with DB-37 connectors. Once properly installed, these should not normally be disturbed. The Network Emulator Unit has a number of jumpers that must be properly wired-up depending on the type of network one wants - a bus, a ring, or a star. This must correspond to the setting of the Topology selector switch. The Network Emulator Unit also has selector switches for Data Rate, Bit Delay, Error Rate and Frame Error Rate. These should be set in accordance with the instructions of each experiment. In addition, there is a Reset switch, which should be used before every experiment, and in case of trouble.

# CONTENTS

## **Lab 1. Study of Ethernet LAN Protocol CSMA/CD**

- 1.1 Introduction
- 1.2 Hardware Requirement
- 1.3 Background
- 1.4 Pre lab
- 1.5 Design
- 1.6 Lab Procedure
- 1.7 Post lab

## **Lab 2: Study of Token Ring and Token Bus Protocols**

- 2.1 Introduction
- 2.2 Hardware Requirement
- 2.3 Background
- 2.4 Pre lab
- 2.5 Design
- 2.6 Lab Procedure
  - 2.6.1 Token Bus
  - 2.6.2 Token Ring
- 2.7 Post lab

## **Lab 3: Study of Wireless LAN protocol CSMA/CA**

- 3.1 Introduction
- 3.2 Hardware Requirement
- 3.3 Background
- 3.4 Pre lab
- 3.5 Design
- 3.6 Lab Procedure
- 3.7 Post lab

## **Lab 4: Implementation and study of Stop and Wait Protocols**

- 4.1 Introduction
- 4.2 Hardware Requirement
- 4.3 Background
- 4.4 Pre lab
- 4.5 Design
- 4.6 Lab Procedure
- 4.7 Post lab

## **Lab 5: Implementation and study of Go back N and Selective Repeat Protocol**

- 5.1 Introduction
- 5.2 Hardware Requirement
- 5.3 Background
- 5.4 Pre lab
- 5.5 Design

## 5.6 Lab Procedure

### 5.6.1 Go Back N protocol

### 5.6.2 Selective Repeat Protocol

## 5.7 Post lab

## **Lab 6: Implementation of Distance Vector Routing algorithm**

### 6.1 Introduction

### 6.2 Hardware Requirement

### 6.3 Background

### 6.4 Pre lab

### 6.5 Design

### 6.6 Lab Procedure

### 6.7 Post lab

## **Lab 7: Implementation of Link State Routing Algorithm**

### 7.1 Introduction

### 7.2 Hardware Requirement

### 7.3 Background

### 7.4 Pre lab

### 7.5 Design

### 7.6 Lab Procedure

### 7.7 Post lab

## **Lab8: Implementation of data encryption and decryption**

### 8.1 Introduction

### 8.2 Hardware Requirement

### 8.3 Background

### 8.4 Pre lab

### 8.5 Design

### 8.6 Lab Procedure

### 8.7 Post lab

## **Lab9: Transfer of files from PC to PC using Windows/ UNIX socket processing**

### 9.1 Introduction

### 9.2 Hardware Requirement

### 9.3 Background

### 9.4 Pre lab

### 9.5 Design

### 9.6 Lab Procedure

### 9.7 Post lab

## **Appendix**

# Lab 1: Study of Ethernet LAN Protocol CSMA/CD

## 1.1 Introduction:

The purpose of this experiment is to understand the concept of Listen-while-transmit to improve efficiency. In this lab, you will be able to implement the CSMA/CD protocol for packet communication between a number of nodes connected to a common bus.

## 1.2 Hardware Requirement

- 3PCs with NIU card
- Network Emulation Unit
- Jumper Cables

## 1.3 Background

*Carrier Sense Multiple Access / Collision Detection*, a set of rules determining how network devices respond when two devices attempt to use a data channel simultaneously (called a *collision*). Standard Ethernet networks use CSMA/CD to physically monitor the traffic on the line at participating stations. To translate this into Ethernet terms, each interface must wait until there is no signal on the channel, then it can begin transmitting. If some other interface is transmitting there will be a signal on the channel, which is called carrier. All other interfaces must wait until carrier ceases before trying to transmit, and this process is called Carrier Sense.

If no transmission is taking place at the time, the particular station can transmit. If two stations attempt to transmit simultaneously, this causes a collision, which is detected by all participating stations. After a random time interval, the stations that collided attempt to transmit again. If another collision occurs, the time intervals from which the random waiting time is selected are increased step by step. This is known as exponential back off.

## 1.4 Pre lab questions

1. Define network topology and list the types with the advantages and disadvantages.
2. Which media access method is used in Ethernet?
3. Which OSI Layer includes the CSMA/CD mechanism?
4. How is the collision occurrence intimated to other nodes in a network?





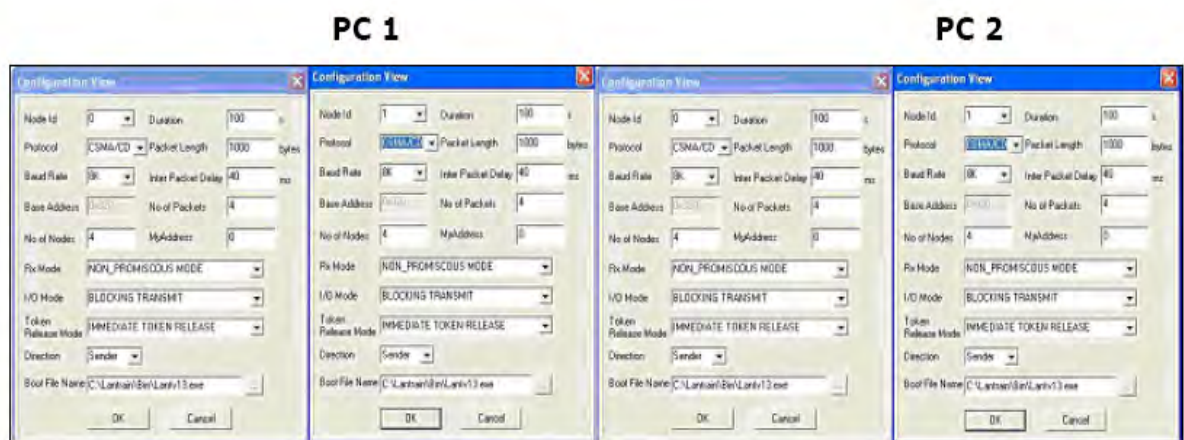
## 5. Define throughput.

### 1.5 Design

Design a network with four logical stations generating a packet length of 100 bytes and run the simulation for 100 seconds. Comment the performance using throughput measures.

### 1.6 Procedure:

1. Click on the **MAC** Experiment icon  twice from the desktop on both PC's.
2. Click the Configuration button  in the window in both the PC's.



### Configuration menu:

PC 1		PC 2	
<b>Node id</b>	0 on config menu 1 and 1 on config menu 2	<b>Node id</b>	0 on config menu 1 and 1 on config menu 2
<b>Protocol</b>	CSMA/CD	<b>Protocol</b>	CSMA/CD
<b>Baud Rate</b>	8Kbps (At both the config menu and NEU)	<b>Baud Rate</b>	8Kbps (At both the config menu and NEU)
<b>Duration</b>	100s	<b>Duration</b>	100s
<b>Packet</b>	1000 bytes	<b>Packet</b>	1000 bytes
<b>Bit Delay</b>	0(at NEU)	<b>Bit Delay</b>	0(at NEU)
<b>Direction</b>	Sender	<b>Direction</b>	Sender

Note: All the nodes have to be configured as 'Senders'. Set the topology as 'Bus'.

$$G = \frac{N * P}{C * t_a} \rightarrow \text{Equation A}$$

**G** is the generated load in the network.


**N** is the number of nodes participating in the network. For example, let us say that 4 nodes (using 2 computers)

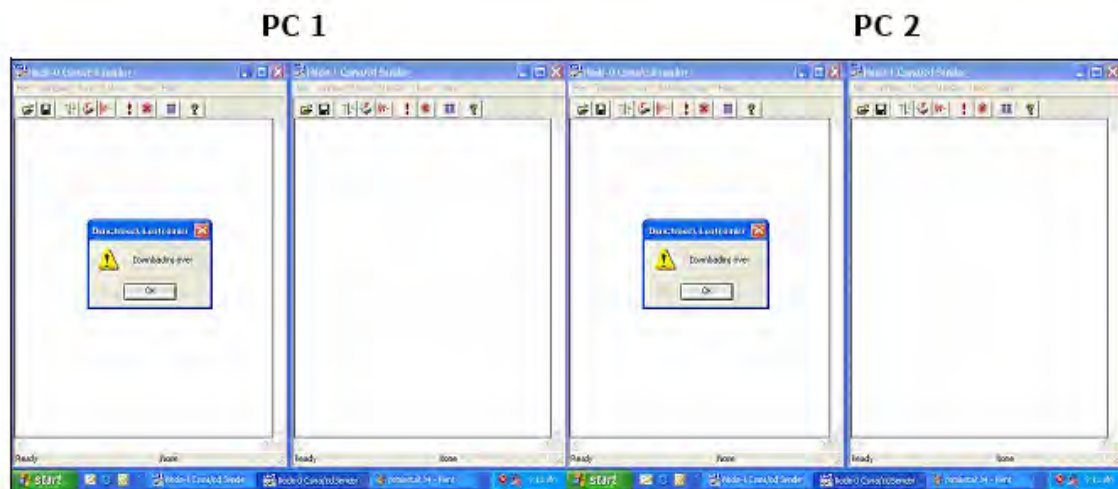
**P** is the packet length expressed in bits; say 100 bytes (800 bits).



**C** is the data rate normally set as 8kbs, which is selected in the NEU.

**t<sub>a</sub>** is the inter packet delay expressed in seconds; the time interval between two consecutive packets generated.

So, let's assume **t<sub>a</sub>** = 40 milliseconds and substitute the above mentioned parameters in the Equation A which leads to **G** = 10. Like wise assume various values of **t<sub>a</sub>** to generate offer loads in the range of 0.1 to 10. Substitute the value of **t<sub>a</sub>** in the configuration menu.

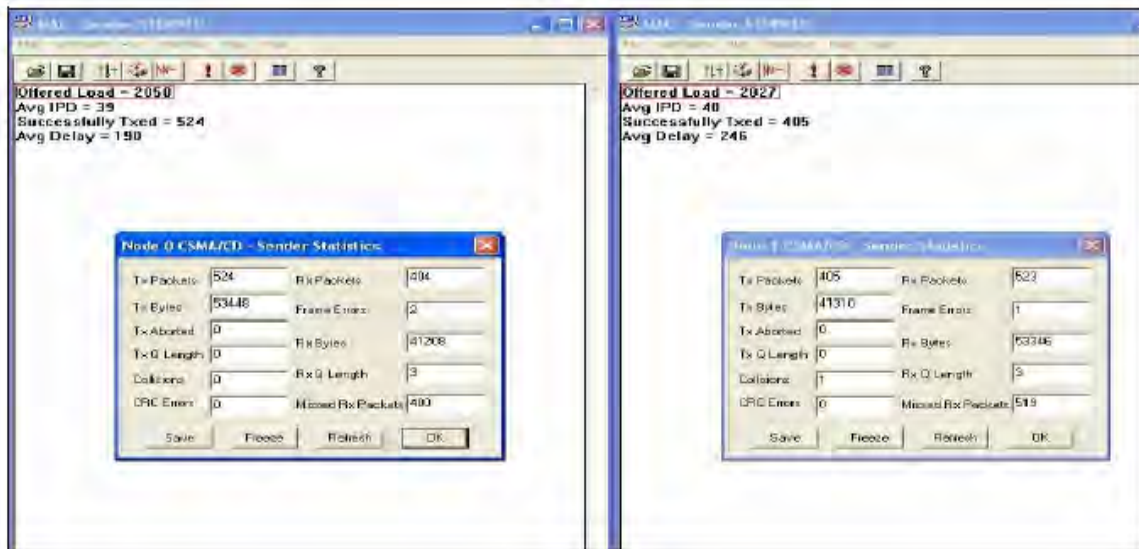
3. Click **OK** button and Download the driver to the NIU using the BOOT button  command. Booting from any one of the applications is enough.



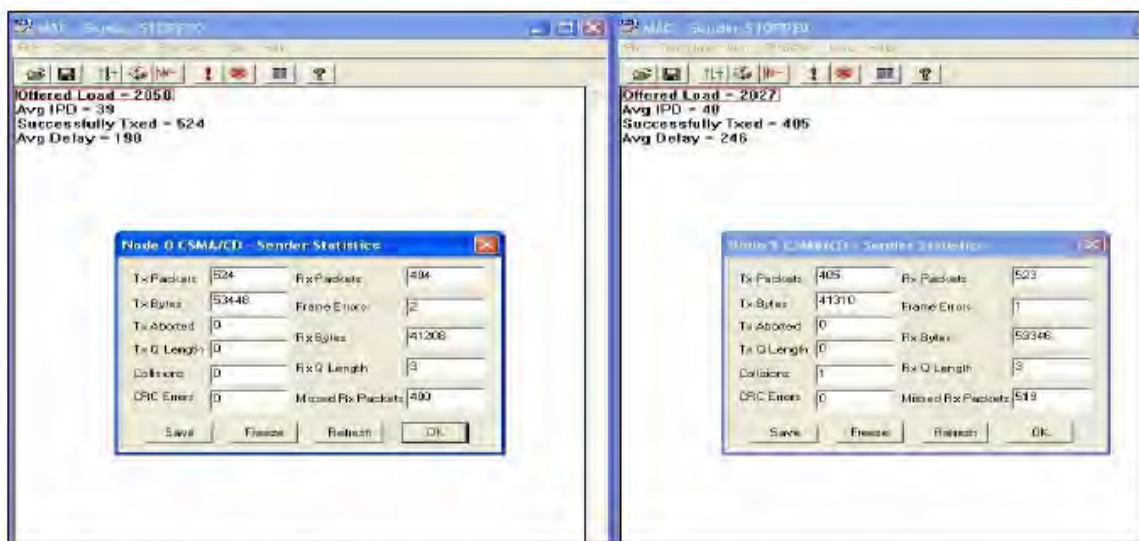
4. Run the experiment by clicking button  or by choosing RUN → Start from each application.
5. View the statistics window for results. To view the statistics window click on  button. Only Tx packets and successfully transmitted packets are taken into account for CSMA/CD calculation.

$$X = \frac{(\text{Sum of Tx packet in all the nodes} * \text{Packet Length} * 8)}{(\text{Duration of Experiment} * \text{Data rate})}$$

### PC 1



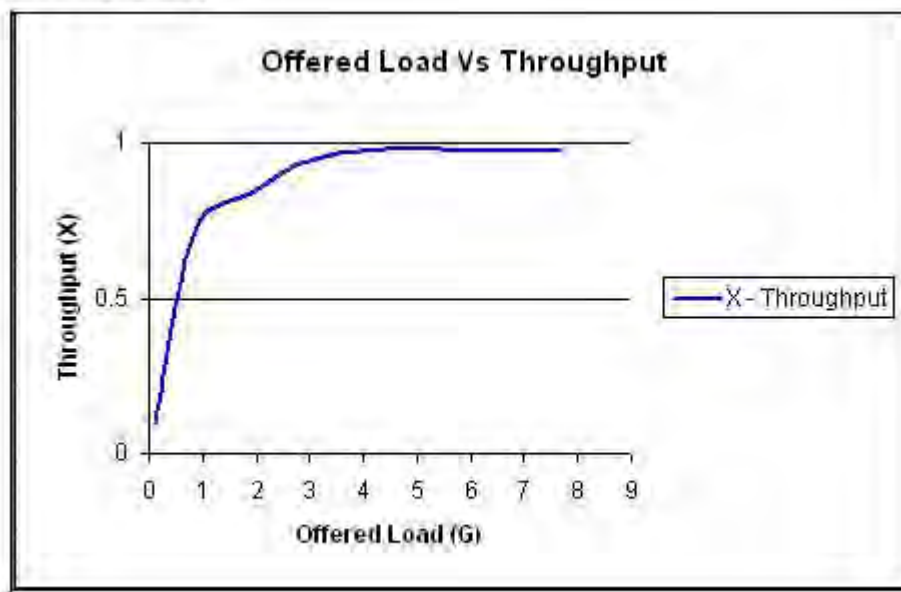
### PC 2



- Note down the readings once the experiment is completed.
  - Repeat the above steps for various values of  $t_a$ .
  - Calculate the Practical offered load from the below given formula and plot the graph between the practical Offered load and Throughput.
- Note:** You can also use the template for plotting the graph. Please refer *Appendix-1* to plot the graph using the template.
- Repeat the experiment for various values of Packet length, Node, Data rate.

**Model tabulation:**

IPD	Tx1	Tx 2	Tx 3	Tx 4	G – Practical Offered Load	X – Practical Throughput
4000	27	25	24	24	0.101	0.1
2000	49	49	49	49	0.208	0.196
800	118	116	119	116	0.5	0.469
400	187	189	191	191	0.976	0.758
200	117	133	290	300	1.904	0.84
100	472	463	13	18	3.555	0.966
40	466	458	20	28	7.687	0.972

**Model Graph:****1.7 Post lab questions:**

1. Calculate the throughput for a network with bandwidth 10 Mbps which can pass at an average of 12,000 frames /min each carrying 10,000 bits.
2. What are the drawbacks of CSMA/CD protocol?
3. Can a node transmit packets during back-off period? Why?
4. How to overcome the drawbacks of CSMA/CD?

## Lab 2: Study of Token Ring and Token Bus protocol

### 2.1 Introduction:

The purpose of this experiment is to understand the concept of demand assignment versus random access, setting priorities and token management in a ring and bus LAN .In this lab you will be able to implement a token –passing access method for a ring and bus LAN.

### 2.2 Hardware Requirement

- 3PCs with NIU card
- Network Emulation Unit
- Jumper Cables

### 2.3 Background

**Token ring** local area network (LAN) technology is a local area network protocol which resides at the data link layer (DLL) of the OSI model. It uses a special three-byte frame called a token that travels around the ring. Token ring frames travel completely around the loop. Stations on a token ring LAN are logically organized in a ring topology with data being transmitted sequentially from one ring station to the next with a control token circulating around the ring controlling access..Physically, a token ring network is wired as a star, with 'hubs' and arms out to each station and the loop going out-and-back through each.

Each station passes or *repeats* the special token frame around the ring to its nearest downstream neighbor. This token-passing process is used to arbitrate access to the shared ring media. Stations that have data frames to transmit must first acquire the token before they can transmit them. Token ring LANs normally use differential Manchester encoding of bits on the LAN media.

When no station is transmitting a data frame, a special token frame circles the loop. This special token frame is repeated from station to station until arriving at a station that needs to transmit data. When a station needs to transmit data, it converts the token frame into a data frame for transmission. Once the sending station receives its own data frame, it converts the frame back into a token

**Token bus** is a network implementing the token ring protocol over a "virtual ring" on a coaxial cable. A token is passed around the network nodes and only the node possessing the token may transmit. If a node doesn't have anything to send, the token is passed on to the next



node on the virtual ring. Each node must know the address of its neighbour in the ring, so a special protocol is needed to notify the other nodes of connections to, and disconnections from, the ring.

## 2.4 Pre Lab Questions

1. What is meant by backbone in a network?
2. What is meant by token and how is it used in ring methodology?
3. Give the advantages of token ring over Ethernet.
4. Define Token Holding Timer (THT).
5. Give the IEEE standard of token ring and token bus LAN.

## 2.5 Design

Design a ring and bus topology with 4 nodes using token passing mechanism and compare the throughput. Set the lowest priority node with My Address 3 and token holding time of 10000 ms. Ensure Bit delay of 0 seconds is set at NEU.

## 2.6 Procedure

### 2.6.1 Token Bus



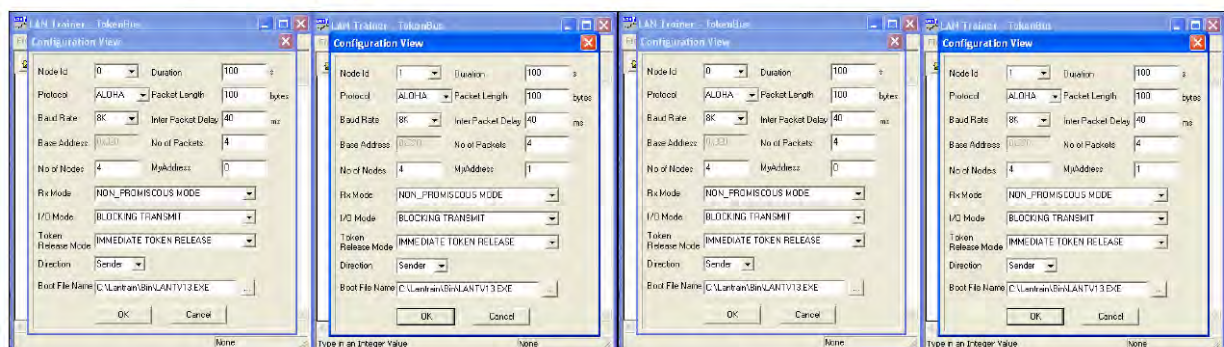
1. Click on the Token Bus icon twice from the desktop.



2. Click the Configuration button in the window in both the PC's.

**PC 1 – Sender**

**PC – 2 Sender**



Setting the Configurations Menu for Token Bus

PC 1		PC 2	
<b>Node id</b>	0 on config menu 1 and 1 on config menu 2	<b>Node id</b>	0 on config menu 1 and 1 on config menu 2
<b>Protocol</b>	ALOHA	<b>Protocol</b>	ALOHA
<b>Baud Rate</b>	8Kbps (At both the config menu and NEU)	<b>Baud Rate</b>	8Kbps (At both the config menu and NEU)
<b>Duration</b>	100s	<b>Duration</b>	100s
<b>Packet Length</b>	100 bytes	<b>Packet Length</b>	100 bytes
<b>My Address</b>	0 on config menu 1 and 1 on menu 2	<b>My Address</b>	2 on config menu 1 and 3 on menu 2
<b>Bit Delay</b>	0(at NEU)	<b>Bit Delay</b>	0(at NEU)
<b>Direction</b>	Sender	<b>Direction</b>	Sender

1.If you connect two PC's and configured four nodes then set the My Address as 0 to 3 in all four nodes, if you connect three PCs and configured six nodes then set the My Address as 0 to 5 in all six nodes.

2.Start running the experiment from the lowest priority node (i.e., from My Address 3 in case of four nodes and 5 in the case of six nodes)

3. No of Nodes has to be set as 4 when two PCs are connected and 6 when three PCs are connected.

$$G = \frac{N * P}{C * t_a} \rightarrow \text{Equation A}$$

G is the generated load in the network.

N is the number of nodes participating in the network. For example, let us say that 4 nodes (using 2 computers)

P is the packet length expressed in bits; say 100 bytes (800 bits).

C is the data rate normally set as 8kbs, which is selected in the NEU.

$t_a$  is the inter packet delay expressed in seconds; the time interval between two consecutive packets generated.

So, lets assume  $t_a = 40$  milliseconds and substitute the above mentioned parameters in the Equation A which leads to  $G = 10$ . Like wise assume various values of  $t_a$  to generate offer loads in the range of 0.1 to 10. Substitute the value of  $t_a$  in the configuration menu.

3. Click OK button and Download the driver to the NIU using the BOOT button



command.

Booting from any one of the applications is enough.



4. Run the experiment by clicking button or by choosing RUN \_ Start from each application.

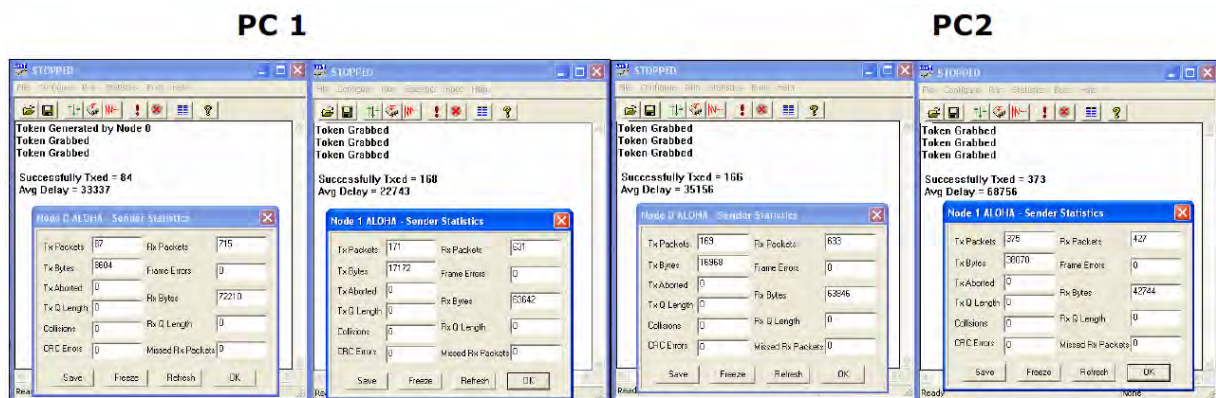
Run the all the experiments at the same time.

Note: While you do this THT window pops up, enter the THT time in all nodes and press the OK

button first in the node, which has the lowest priority of My Address.

5. Set the Token Holding Time (THT) (say 10000 ms).

6. View the statistics window for results. To view the statistics window click on button.



7. Note down the readings once the experiment is completed.

8. Repeat the above steps for various values of  $t_a$

9. Calculate the Practical offered load from the below given formula and plot the graph between the practical Offered load and Throughput.

Note: You can also use the template for plotting the graph. Please refer Appendix-1 to plot the graph using the template.

10. Repeat the experiment for various values of Packet length, Node, Data rate.

11. Repeat the above steps, while running the experiment set the BER to 10-2 in the NEU or try to stop one of the nodes and observe the behavior and explain the same.

Calculations of Practical Throughput from the obtained readings



$$X = \frac{(\text{Sum of Tx packet in all the nodes} * \text{Packet Length} * 8)}{(\text{Duration of Experiment} * \text{Data rate})}$$

Calculations of Offered Load

$$G = \frac{N * P}{C * t_a}$$

G – Offered load

N – Number of nodes

P – Packet length in bits

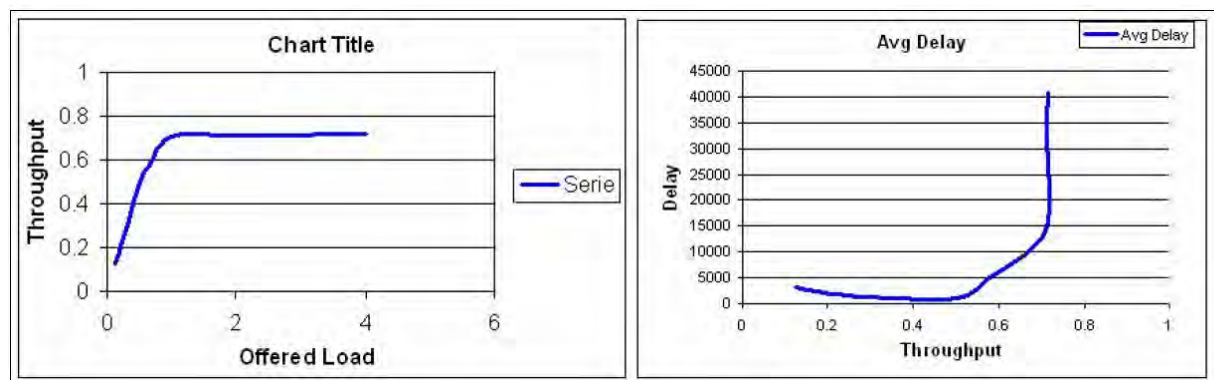
C – Data rate in bits/sec

t<sub>a</sub> -Inter packet delay in millisecs.

Model Tabulations

IPD	Tx 1	Tx 2	Tx 3	Tx 4	G - Offered Load	X –Practical Throughput	Avg Delay
16000	7	7	9	5	0.25	0.28	12205.5
8000	11	11	11	13	0.5	0.46	7787.75
4000	18	21	26	19	1	0.84	11265
2000	24	20	21	24	2	0.89	28604.75
1000	20	18	30	20	4	0.88	38216

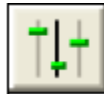
Model Graph



## 2.6.2 Token Ring



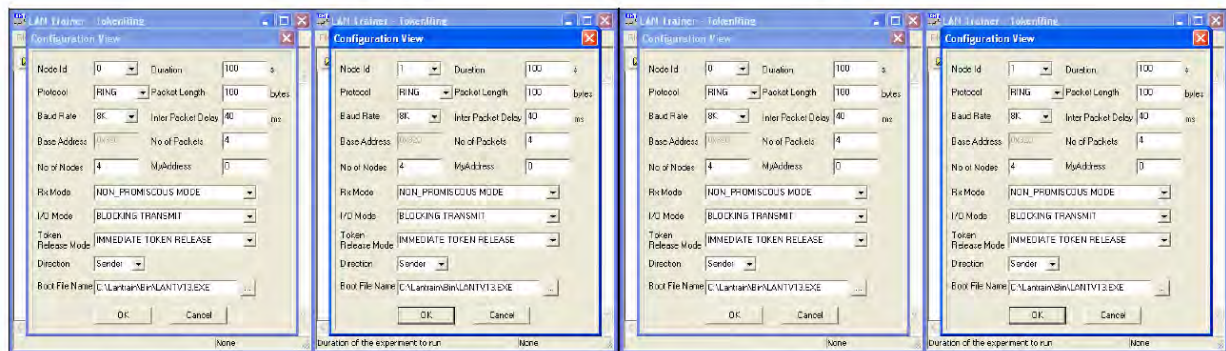
1. Click on the Token ring icon twice from the desktop.



2. Click the Configuration button in the window in both the PC's.

**PC-1**

**PC-2**



Setting the configuration menu:

PC 1		PC 2	
<b>Node id</b>	0 on config menu 1 and 1 on config menu 2	<b>Node id</b>	0 on config menu 1 and 1 on config menu 2
<b>Protocol</b>	RING	<b>Protocol</b>	RING
<b>Baud Rate</b>	8Kbps (At both the config menu and NEU)	<b>Baud Rate</b>	8Kbps (At both the config menu and NEU)
<b>Duration</b>	100s	<b>Duration</b>	100s
<b>Packet Length</b>	1000 bytes	<b>Packet Length</b>	1000 bytes
<b>Bit Delay</b>	0(at NEU)	<b>Bit Delay</b>	0(at NEU)
<b>Direction</b>	Sender	<b>Direction</b>	Sender

$$G = \frac{N * P}{C * t_a}$$

→ Equation A

G is the generated load in the network.

N is the number of nodes participating in the network. For example, let us say that 4 nodes (using 2 computers)


P is the packet length expressed in bits; say 100 bytes (800 bits).

C is the data rate normally set as 8kbs, which is selected in the NEU.

ta is the inter packet delay expressed in seconds; the time interval between two consecutive

packets generated.

So, let's assume  $t_a = 40$  milliseconds and substitute the above mentioned parameters in the Equation A which leads to  $G = 10$ . Like wise assume various values of  $t_a$  to generate offer loads in the range of 0.1 to 10. Substitute the value of  $t_a$  in the configuration menu.

3. Click OK button and Download the driver to the NIU using the BOOT  button command.


Booting from any one of the applications is enough.

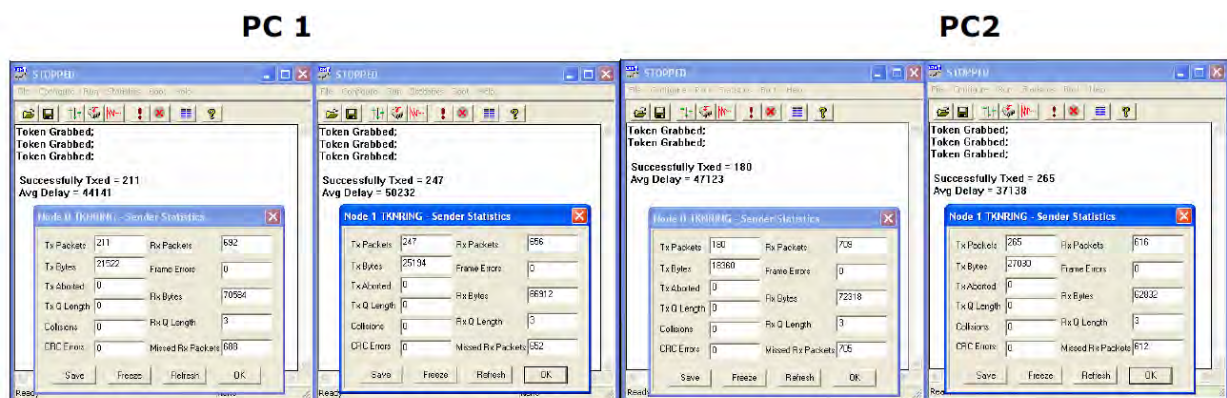
4. Run the experiment by clicking button  or by choosing RUN \_ Start from each application.

Run all the experiments at the same time.

5. Set the Token Holding Time (THT) (say 10000 ms).

6. View the statistics window for results. To view the statistics window click on

button  .



7. Note down the readings once the experiment is completed.

8. Repeat the above steps for various values of  $t_a$ .

9. Calculate the Practical offered load from the below given formula and plot the graph between the practical Offered load and Throughput.

Note: You can also use the template for plotting the graph.

10. Repeat the experiments for various values of Packet length, Node, Data rate.

11. Repeat the above steps, while running the experiment set the BER to 10<sup>-2</sup> in the NEU or try

to stop one of the nodes and observe the behavior and explain the same.

Calculation of Practical Throughput (X) from the obtained readings:

$$X = \frac{(\text{Sum of Tx packet in all the nodes} * \text{Packet Length} * 8)}{(\text{Duration of Experiment} * \text{Data rate})}$$

Calculation of the Offered load:

$$G = \frac{N * P}{C * t_a}$$

G – Offered load

N – Number of nodes

P – Packet length in bits

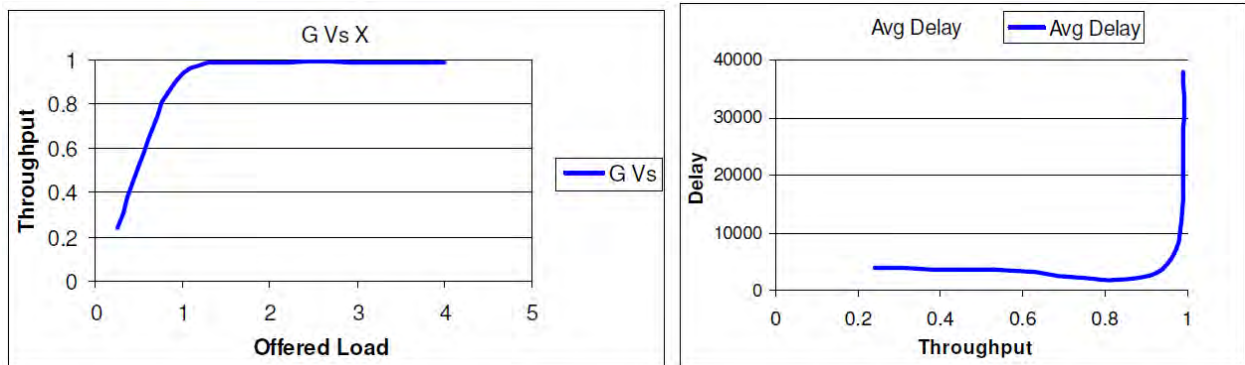
C – Data rate in bits/sec

t<sub>a</sub> – Inter packet delay in millisecs.

Model Tabulation:

IPD	Tx Node1	Tx Node2	Tx Node3	Tx Node4	G - Offered Load	X - Throughput	Avg Delay
16000	6	6	6	6	0.25	0.24	3779.75
8000	12	13	14	14	0.5	0.53	3708.25
4000	19	19	28	28	1	0.94	3658.5
2000	22	22	30	25	2	0.99	28270.75
1000	30	27	21	21	4	0.99	37720.25

## Model Graph



### 2.7 Post lab questions:

1. Calculate the propagation time and transmission time for 2.5Kbyte message if the bandwidth of 1Gbps? distance is 12,000 Km and light travels at  $2.4 \times 10^8$  m/s
2. Give the frame format of Token Ring LAN
3. What type of token passing used in FDDI.
4. Which network architecture uses a dual-ring topology?
5. Compare the performance between ring and bus methodology

## **Lab 3: Study of Wireless LAN protocol CSMA/CA**

### **3.1 Introduction**

The purpose of this experiment is to introduce you the concepts of virtual channel sensing using RTS/CTS, physical medium reservation and RTS-CTS-DATA-ACK cycle. In this lab you will be able to implement the CSMA/CA protocol for packet communication between a number of nodes connected to a common bus.

### **3.2 Hardware Requirement**

- 3PCs with NIU card
- Network Emulation Unit
- Jumper Cables

### **3.3 Background**

**Carrier Sense Multiple Access With Collision Avoidance (CSMA/CA)**, in computer networking, is a wireless network multiple access method in which:

- a carrier sensing scheme is used.
- a node wishing to transmit data has to first listen to the channel for a predetermined amount of time to determine whether or not another node is transmitting on the channel within the wireless range. If the channel is sensed "idle," then the node is permitted to begin the transmission process. If the channel is sensed as "busy," the node defers its transmission for a random period of time. Once the transmission process begins, it is still possible for the actual transmission of application data to not occur.

**CSMA/CA** is a modification of pure Carrier Sense Multiple Access (**CSMA**). Collision avoidance is used to improve **CSMA** performance by not allowing wireless transmission of a node if another node is transmitting, thus reducing the probability of collision due to the use of a random time. Optionally, but almost always implemented, an IEEE 802.11 RTS/CTS exchange can be required to better handle situations such as the hidden node problem in wireless networking.

CSMA/CA is a layer 2 access method, not a protocol of the OSI model. A node wishing to send data initiates the process by sending a Request to Send frame (RTS). The destination node replies with a Clear to send frame (CTS). Any other node receiving the RTS or CTS frame should refrain from sending data for a given time (solving the hidden node problem). The amount of time the node should wait before trying to get access to the medium is included in both the RTS and the CTS frame. This protocol was designed under the assumption that all nodes have the same transmission range.

RTS/CTS is an additional method to implement *virtual carrier sensing* in Carrier sense multiple access with collision avoidance (CSMA/CA). By default, 802.11 relies on *physical carrier sensing* only which is known to suffer from the hidden terminal problem.

RTS/CTS packet size threshold is 0-2347 octets. Typically, sending RTS/CTS frames does not occur unless the packet size exceeds this threshold. If the packet size the node wants to transmit is larger than the threshold, the RTS/CTS handshake gets triggered. Otherwise, the data frame gets sent immediately. RTS/CTS packets carry the expected duration of the data transmission, which will have some implications.

### **3.4 Pre lab questions**

1. What is the difference between CSMA/CD and CSMA/CA?
2. What are hidden terminal problem and exposed terminal problem?
3. What is DCF?
4. What is the use of RTS/CTS handshake protocol?
5. What is random back off algorithm?



### **3.5 Design**

Design a network with two senders and one receiver and show how collision is reduced when two users try to access the channel to send data to same receiver.

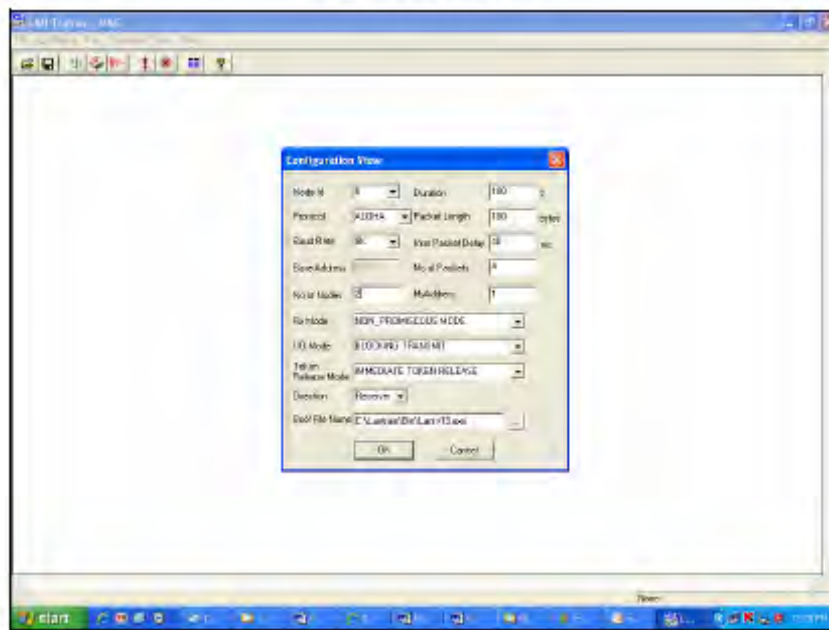


### 3.6 Procedure

### Steps to configure Receiver:

1. Click on the CSMA/CA icon  from the desktop on one PC.
2. Click the Configuration  button.

## PC 1 Receiver





Configuration menu for Access point:

PC 1	
Node id	0
Protocol	ALOHA
Baud Rate	8Kbps (At both the config menu and NEU)
Duration	100 s
Packet Length	100
Bit Delay	0 (at NEU)
Direction	Receiver
My Address	1

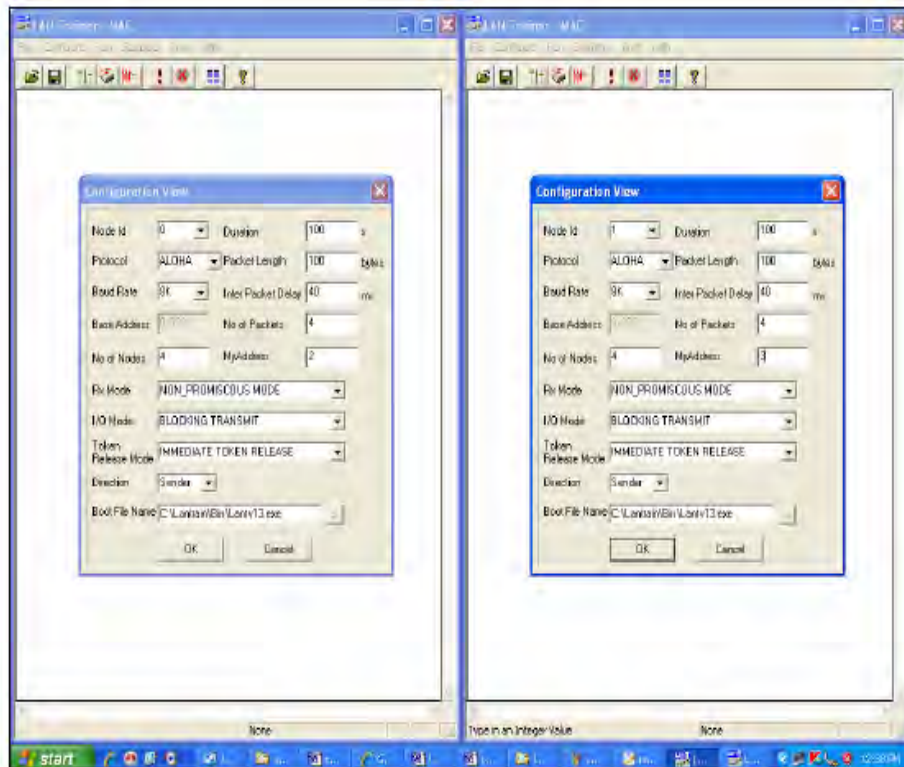
Note: There should be only one Access Point and no clients in PC1. Set the topology as BUS.



### Steps to configure Clients nodes:

- Click on the **CSMACA** icon  from the desktop on the second PC.
- Click the Configuration  button in the window in the second PC.

### PC 2



### Setting the configuration menu for client nodes:

PC 2	
<b>Node id</b>	0 on config menu 1 and 1 on config menu 2
<b>Protocol</b>	ALOHA
<b>Baud Rate</b>	8Kbps (At both the config menu and NEU)
<b>Duration</b>	100 s
<b>Packet Length</b>	100
<b>Bit Delay</b>	0 (at NEU)
<b>Direction</b>	Sender
<b>My Address</b>	2 on config menu 1 and 3 on config menu 2

**Note:** Here the My Address should be 2 for Node Id 0 and should be 3 for Node id 1. If PC3 is used, then the My Address should be 4 for Node Id 0 and should be 5 for Node id 1 on that PC.

$$G = \frac{N * P}{C * t_a}$$

→ Equation A

**G** is the generated load in the network.


**N** is the number of nodes participating in the network (for CSMA/CA **N** is the number of clients). For example, let us say that 2 nodes (using 1 computer; 4 nodes using 2 computers). In CSMA/CA the access point is considered as load generating node.


**P** is the packet length expressed in bits; say 100 bytes (800 bits).

**C** is the data rate normally set as 8kbs, which is selected in the NEU.

**t<sub>a</sub>** is the inter packet delay expressed in seconds; the time interval between two consecutive packets generated.

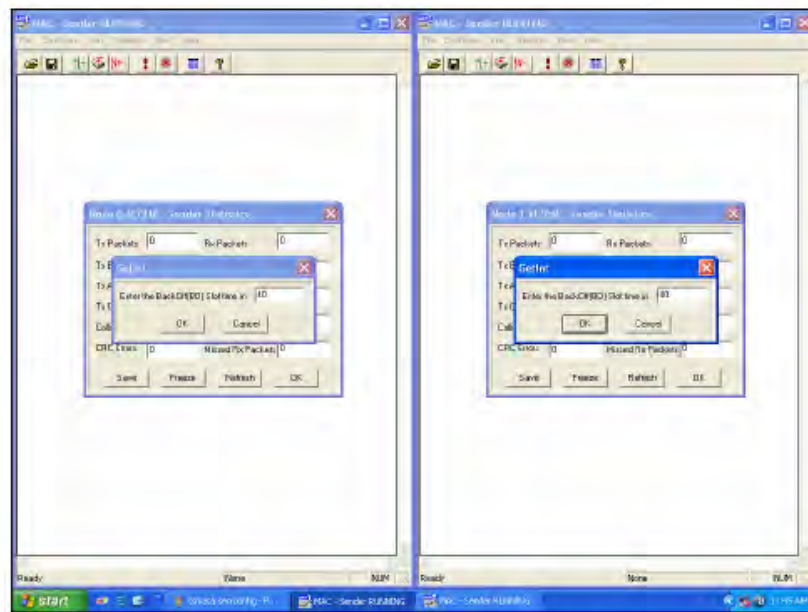
So, let's assume **t<sub>a</sub>** = 40 milliseconds and substitute the above mentioned parameters in the Equation A which leads to **G** = 5. Like wise assume various values of **t<sub>a</sub>** to generate offer loads in the range of 0.1 to 10. Substitute the value of **t<sub>a</sub>** in the configuration menu.

5. Click **OK** button and Download the driver to the NIU using the BOOT button  command. Booting from any one of the applications is enough.

6. Run the experiment by clicking button  or by choosing RUN → Start from each application. Follow the same procedures in the client side also. Once when you press the run button the following window will appear.

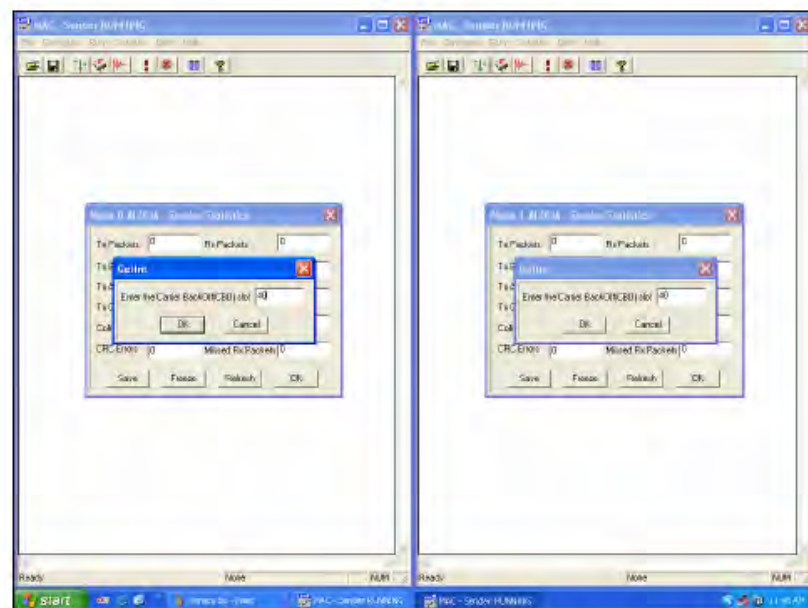
7. Set the **Back-off time** as 40 ms simultaneously in both the windows.

## PC2



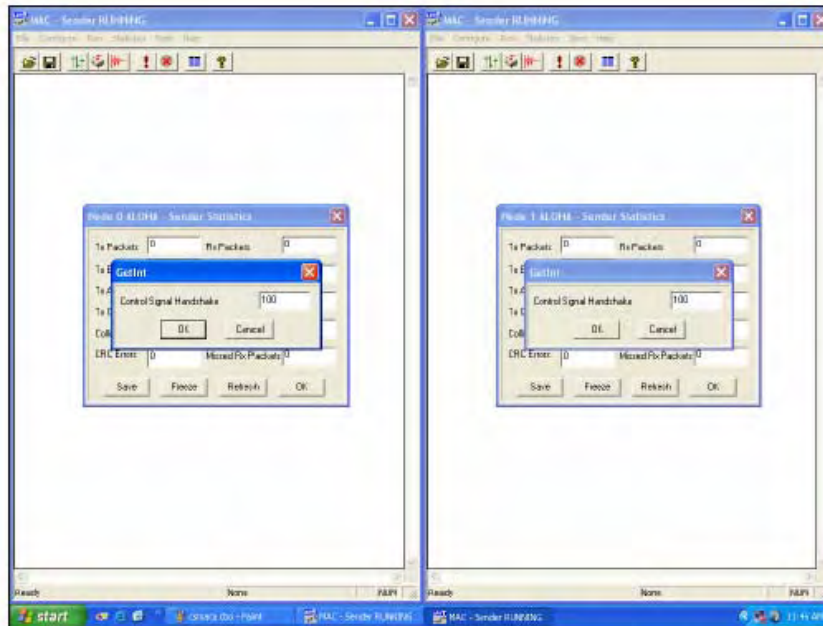
- Set the **Carrier Back-off time** as 40 ms simultaneously in both the windows.

## PC 2



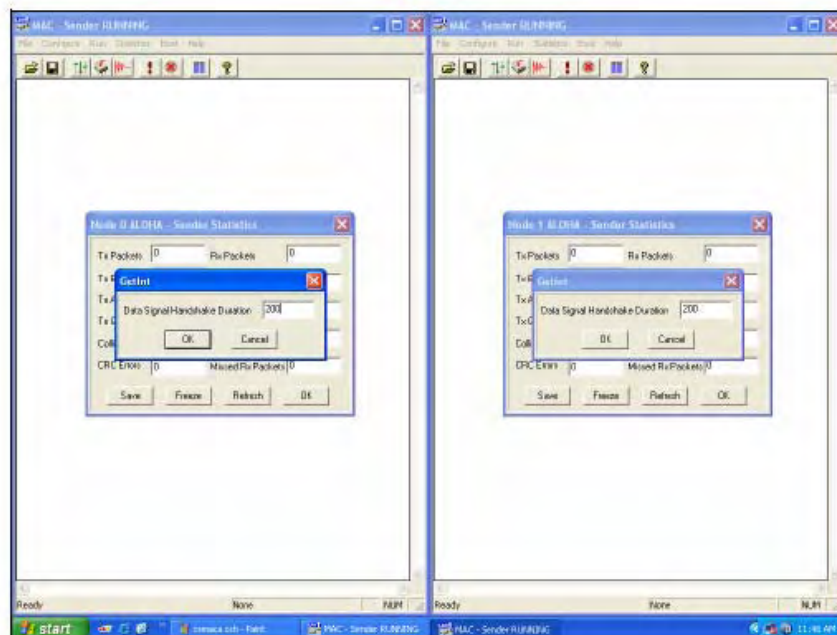
- Set the **Control Signal Handshake** equal to Packet length (for this case it is 100 as the packet length is 100 bytes) simultaneously in both the windows.

## PC 2



10. Set the **Data Signal Handshake** equal to twice the Packet length (for this case it is 200 as the packet length is 100 bytes) simultaneously in both the windows.

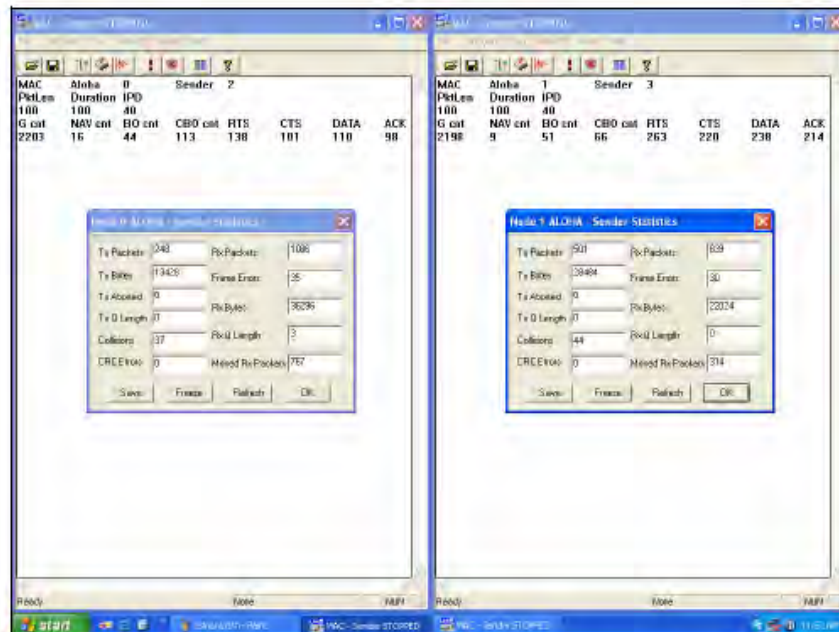
## PC 2






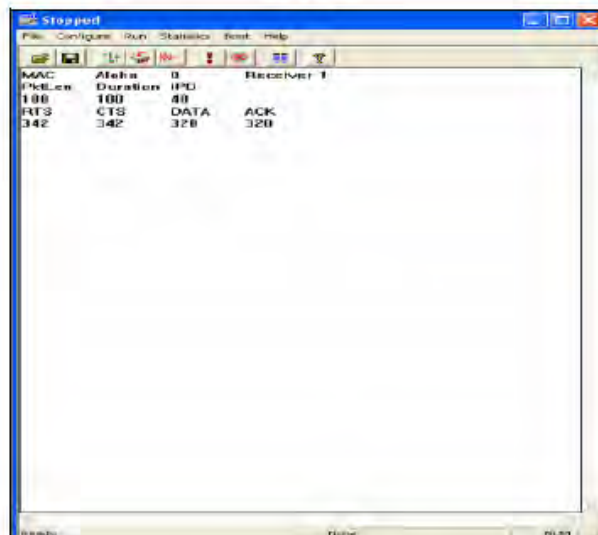
11. After 100 secs both the nodes stops transmitting and the following screen appear.

### PC 2



12. Once the Sender stops press the **OK** button in the receiver Node and Press the **stop**  Buttn and the following screen appears

### PC1



Note down the readings once the experiment is completed.

13. Repeat the above steps for various values of  $t_a$ .

14. Calculate the Practical offered load from the below given formula and plot the graph between the practical Offered load and Throughput.

**Note:** You can also use the template for plotting the graph. Please refer *Appendix-1* to plot the graph using the template.

15. Repeat the experiment for various values of Packet length, Node, Data rate.

**Calculation of Practical Throughput for CSMA/CA:**

$$X = \frac{(\text{Sum of ACK counts in all the clients} * \text{Packet Length} * 8)}{(\text{Duration of Experiment} * \text{Data Rate})}$$

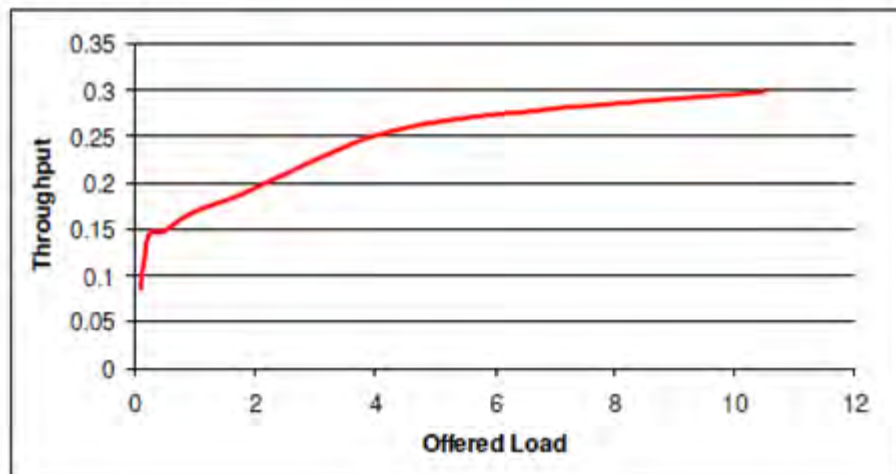
**Calculation of Practical Offered Load for CSMA/CA:**

$$G = \frac{(\text{Sum of G count in all the clients} * \text{Packet Length} * 8)}{(\text{Duration of Experiment} * \text{Data Rate})}$$

### Model Tabulation:

IPD (ms)	G Count1	G count2	Ack count1	Ack count2	Practical G	X-Practical Throughput
2000	49	50	43	39	0.099	0.082
800	120	117	67	76	0.237	0.143
400	255	258	82	91	0.513	0.173
200	470	473	99	83	0.948	0.182
100	925	925	98	91	1.85	0.189
40	2046	2046	160	138	4.092	0.298
20	3467	3468	176	43	6.935	0.319
10	5231	5224	184	151	10.455	0.335

### Model Graph:



### 3.7 Post lab questions

1. What is NAV?
2. What do you mean by exponential back off algorithm?
3. Fragment a 1000 byte packet into five 200 bytes packet and compare the throughput obtained with single 1000 byte packet transmission
4. In what situations can collision occur in WLAN network? How to solve the collisions or minimize the probability of collisions?
5. What is the choice of contention window?

## **Lab 4: Implementation and study of Stop and Wait Protocols**

### **4.1 Introduction:**

The purpose of this experiment is to introduce you to the basics of error correction, time outs and state machines. In this lab, you will be able to provide reliable data transfer between two nodes over an unreliable network using the stop- and-wait protocol.

### **4.2 Hardware Requirement**

- 3PCs with NIU card
- Network Emulation Unit
- Jumper Cables

### **4.3 Background**

Stop and wait is the fundamental technique to provide reliable transfer under unreliable packet delivery system. After transmitting one packet, the sender waits for an acknowledgment (ACK) from the receiver before transmitting the next one. In this way, the sender can recognize that the previous packet is transmitted successfully and we could say "stop-n-wait" guarantees reliable transfer between nodes. To support this feature, the sender keeps a record of each packet it sends. Also, to avoid confusion caused by delayed or duplicated ACKs, "stop-n-wait" send each packet with unique sequence numbers and receive that numbers in each ACK. If the sender doesn't receive ACK for previous sent packet after a certain period of time, the sender times out and retransmits that packet again. There are two cases when the sender doesn't receive ACK; One is when the ACK is lost and the other is when the frame itself is not transmitted. To support this feature, the sender keeps timer per each packet.

### **4.4 Pre lab questions**

1. What are the functions of data link layer?
2. What is the need for flow control techniques?
3. Explain the mechanism of stop and wait flow control technique.
4. In what situations does the sender retransmit a packet?
5. What is meant by promiscuous mode?



## 4.5 Design

Design a network to illustrate reliable data transfer with packet length of 1000 bytes using stop and wait protocol. Illustrate the performance of the network by changing the time out values from 1000 ms to 4000 ms in steps of 500ms.

## 4.6 Procedure:



1. Click on the Stop & Wait icon from the desktop on both PCs.



2. Click the Configuration button in the window in both the Pc's.

### PC 1 SENDER


### PC 2 RECEIVER

Configuration View		Configuration View	
Node Id	0	Node Id	0
Duration	100 s	Duration	100 s
Protocol	CSMA/CD	Protocol	CSMA/CD
Packet Length	1000 bytes	Packet Length	1000 bytes
Baud Rate	8K	Baud Rate	8K
Inter Packet Delay	400 ms	Inter Packet Delay	400 ms
Base Address	0x320	Base Address	0x320
No of Packets	4	No of Packets	4
No of Nodes	4	No of Nodes	4
MyAddress	0	MyAddress	0
Rx Mode	NON_PROMISCOUS MODE	Rx Mode	NON_PROMISCOUS MODE
I/O Mode	BLOCKING TRANSMIT	I/O Mode	BLOCKING TRANSMIT
Token Release Mode	IMMEDIATE TOKEN RELEASE	Token Release Mode	IMMEDIATE TOKEN RELEASE
Direction	Sender	Direction	Receiver
Boot File Name	C:\Lantrain\Bin\Lantv13.exe	Boot File Name	C:\Lantrain\Bin\Lantv13.exe
OK Cancel		OK Cancel	

Setting the configuration menu:

PC 1		PC 2	
<b>Node id</b>	0	<b>Node id</b>	0
<b>Protocol</b>	CSMA/CD	<b>Protocol</b>	CSMA/CD
<b>Baud Rate</b>	8Kbps (At both the config menu and NEU)	<b>Baud Rate</b>	8Kbps (At both the config menu and NEU)
<b>Duration</b>	100s	<b>Duration</b>	100s
<b>Packet Length</b>	1000 bytes	<b>Packet Length</b>	1000 bytes
<b>Bit Delay</b>	0(at NEU)	<b>Bit Delay</b>	0(at NEU)
<b>Direction</b>	Sender	<b>Direction</b>	Receiver

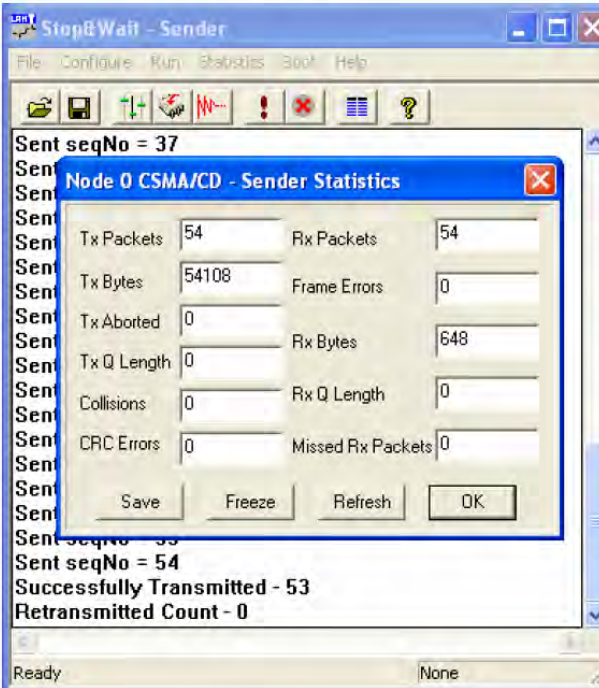
3. Set the Inter Packet Delay to 400msecs

4. Click OK button and Download the driver to the NIU using the BOOT  button command. Booting from any one of the applications is enough.

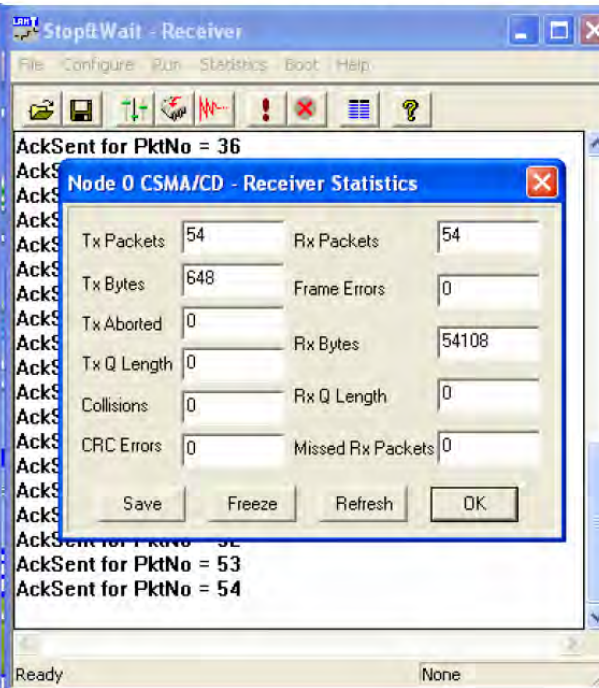
5. Run the experiment by clicking button  or by choosing RUN \_ Start from each application.

6. Set the Timeout Value to 1500 ms

**PC 1 SENDER**



**PC 2 RECEIVER**



7. Note down the no of successfully Transmitted Packets.

8. Repeat the above steps for various time out values and plot the graph between timeout Value &Throughput. Find the optimum timeout value from the plot.

9. Explain why the throughput is less compared to CSMACD protocol.

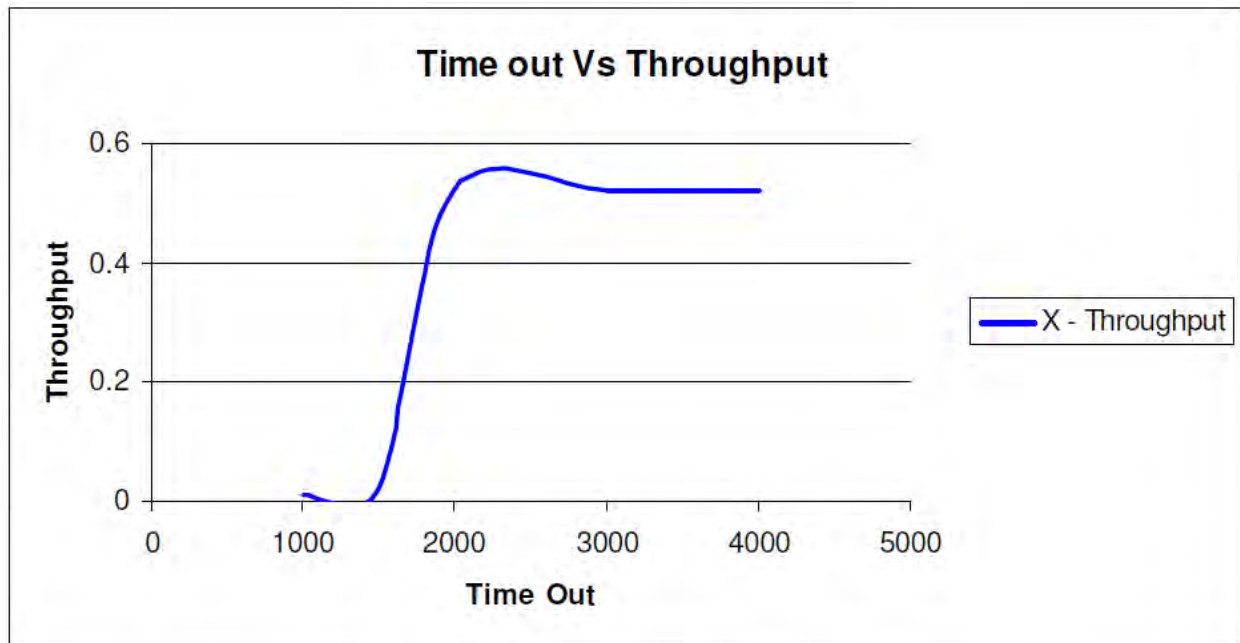
Calculation of Practical Throughput:

$$X = \frac{(\text{Sum of Successfully Tx packets} * \text{Packet Length} * 8)}{(\text{Duration of Experiment} * \text{Data rate})}$$

Model Tabulation:

Time out value in ms	Successfully Tx packets	Practical Throughput
1000	1	0.01
1500	2	0.02
2000	52	0.52
3000	52	0.52
4000	52	0.52

Model Graph:



#### 4.7 Post lab questions

1. Computer A uses stop and wait ARQ protocol to send packets to computer B. If the distance between A and B is 4000km, how long does it take computer A to receive acknowledgement for a packet? Use the speed of light for propagation speed and assume the time between receiving and sending the acknowledgement is zero.
2. Why there is no need to number the acknowledgements in stop and wait ARQ?
3. For the question no1 how long it takes for computer A to send out a packet of size 1000bytes if the throughput is 100,000kbps?

## Lab 5: Implementation and study of Go back N and Selective Repeat Protocol

### 5.1 Introduction:

The purpose of this experiment is to introduce you to the basics of error correction, time outs and state machines. In this lab, you will be able to provide reliable data transfer between two nodes over an unreliable network using Go back N and selective repeat protocol.

### 5.2 Hardware Requirement

- 3PCs with NIU card
- Network Emulation Unit
- Jumper Cables

### 5.3 Background

**Go-Back-N ARQ** is a specific instance of the Automatic Repeat-request (ARQ) Protocol, in which the sending process continues to send a number of frames specified by a *window size* even without receiving an ACK packet from the receiver. It is a special case of the general sliding window protocol with the transmit window size of N and receive window size of 1.

The receiver process keeps track of the sequence number of the next frame it expects to receive, and sends that number with every ACK it sends. The receiver will ignore any frame that does not have the exact sequence number it expects – whether that frame is a "past" duplicate of a frame it has already ACK'ed <sup>[1]</sup> or whether that frame is a "future" frame past the last packet it is waiting for. Once the sender has sent all of the frames in its *window*, it will detect that all of the frames since the first lost frame are *outstanding*, and will go back to sequence number of the last ACK it received from the receiver process and fill its window starting with that frame and continue the process over again.

Go-Back-N ARQ is a more efficient use of a connection than Stop-and-wait ARQ, since unlike waiting for an acknowledgement for each packet, the connection is still being utilized as packets are being sent. In other words, during the time that would otherwise be spent waiting, more packets are being sent. However, this method also results in sending frames

multiple times – if any frame was lost or damaged, or the ACK acknowledging them was lost or damaged, then that frame and all following frames in the window (even if they were received without error) will be re-sent. To avoid this, Selective Repeat ARQ can be used.

**Selective Repeat ARQ / Selective Reject ARQ** is a specific instance of the Automatic Repeat-request (ARQ) Protocol. It may be used as a protocol for the delivery and acknowledgement of message units, or it may be used as a protocol for the delivery of subdivided message sub-units.

When used as the protocol for the delivery of **messages**, the sending process continues to send a number of frames specified by a *window size* even after a frame loss. Unlike Go-Back-N ARQ, the receiving process will continue to accept and acknowledge frames sent after an initial error; this is the general case of the sliding window protocol with both transmit and receive window sizes greater than 1.

The receiver process keeps track of the sequence number of the earliest frame it has not received, and sends that number with every ACK it sends. If a frame from the sender does not reach the receiver, the sender continues to send subsequent frames until it has emptied its *window*. The receiver continues to fill its receiving window with the subsequent frames, replying each time with an ACK containing the sequence number of the earliest missing frame. Once the sender has sent all the frames in its *window*, it re-sends the frame number given by the ACKs, and then continues where it left off.

The size of the sending and receiving windows must be equal, and half the maximum sequence number (assuming that sequence numbers are numbered from 0 to  $n-1$ ) to avoid miscommunication in all cases of packets being dropped. To understand this, consider the case when all ACKs are destroyed. If the receiving window is larger than half the maximum sequence number, some, possibly even all, of the packages that are resent after timeouts are duplicates that are not recognized as such. The sender moves its window for every packet that is acknowledged.

When used as the protocol for the delivery of subdivided messages it works somewhat differently. In non-continuous channels where messages may be variable in length, standard ARQ or Hybrid ARQ protocols may treat the message as a single unit. Alternately selective

retransmission may be employed in conjunction with the basic ARQ mechanism where the message is first subdivided into sub-blocks (typically of fixed length) in a process called Packet segmentation. The original variable length message is thus represented as a concatenation of a variable number of sub-blocks. While in standard ARQ the message as a whole is either acknowledged (ACKed) or negatively acknowledged (NAKed), in ARQ with selective transmission the NAKed response would additionally carry a bit flag indicating the identity of each sub-block successfully received. In ARQ with selective retransmission of sub-divided messages each retransmission diminishes in length, needing to only contain the sub-blocks that were NAKed.

In most channel models with variable length messages, the probability of error-free reception diminishes in inverse proportion with increasing message length. In other words it's easier to receive a short message than a longer message. Therefore standard ARQ techniques involving variable length messages have increased difficulty delivering longer messages, as each repeat is the full length. Selective retransmission applied to variable length messages completely eliminates the difficulty in delivering longer messages, as successfully delivered sub-blocks are retained after each transmission, and the number of outstanding sub-blocks in following transmissions diminishes.

#### **5.4 Pre lab questions**

1. Explain the mechanism of Go back N ARQ.
2. In sliding window flow control if the window size is 63 what is the range of sequence numbers?
3. For a sliding window of size  $n-1$  ( $n$ -sequence numbers) there can be a maximum of ----- frames sent but unacknowledged.
4. Differentiate between bit rate and baud rate.
5. Explain the mechanism of selective repeat ARQ.
6. Compare selective repeat and Goback N ARQ schemes.



## 5.5 Design

Design a network topology to illustrate Go Back N and Selective Repeat Protocols. Assume the window size and inter packet delay as 4 and 400ms respectively. Run the simulation for 100 seconds duration and analyze the throughput performance for various time out values.

## 5.6 Procedure:

### 5.6.1 Go Back N Protocol



1. Click on the Sliding Window GBN icon from the desktop on both PCs.



2. Click the Configuration button in the window in both the Pc's.


PC 1 SENDER	PC 2 RECEIVER
<b>Configuration View</b>	<b>Configuration View</b>
Node Id: 0	Node Id: 0
Duration: 100 s	Duration: 100 s
Protocol: CSMA/CD	Protocol: CSMA/CD
Packet Length: 1000 bytes	Packet Length: 1000 bytes
Baud Rate: 8K	Baud Rate: 8K
Inter Packet Delay: 400 ms	Inter Packet Delay: 400 ms
Base Address: 0x320	Base Address: 0x320
No of Packets: 4	No of Packets: 4
No of Nodes: 4	No of Nodes: 4
MyAddress: 0	MyAddress: 0
Rx Mode: NON_PROMISCUOUS MODE	Rx Mode: NON_PROMISCUOUS MODE
I/O Mode: BLOCKING TRANSMIT	I/O Mode: BLOCKING TRANSMIT
Token Release Mode: IMMEDIATE TOKEN RELEASE	Token Release Mode: IMMEDIATE TOKEN RELEASE
Direction: Sender	Direction: Receiver
Boot File Name: C:\Lantrain\Bin\Lantv13.exe	Boot File Name: C:\Lantrain\Bin\Lantv13.exe
OK Cancel	OK Cancel

Setting the configuration menu:

PC 1		PC 2	
<b>Node id</b>	0	<b>Node id</b>	0
<b>Protocol</b>	CSMA/CD	<b>Protocol</b>	CSMA/CD
<b>Baud Rate</b>	8Kbps (At both the config menu and NEU)	<b>Baud Rate</b>	8Kbps (At both the config menu and NEU)
<b>Duration</b>	100s	<b>Duration</b>	100s
<b>Packet Length</b>	1000 bytes	<b>Packet Length</b>	1000 bytes
<b>Bit Delay</b>	0(at NEU)	<b>Bit Delay</b>	0(at NEU)
<b>Direction</b>	Sender	<b>Direction</b>	Receiver
<b>No of packets</b>	4	<b>No of packets</b>	4

Note: The No of Packets parameter defines the window size.

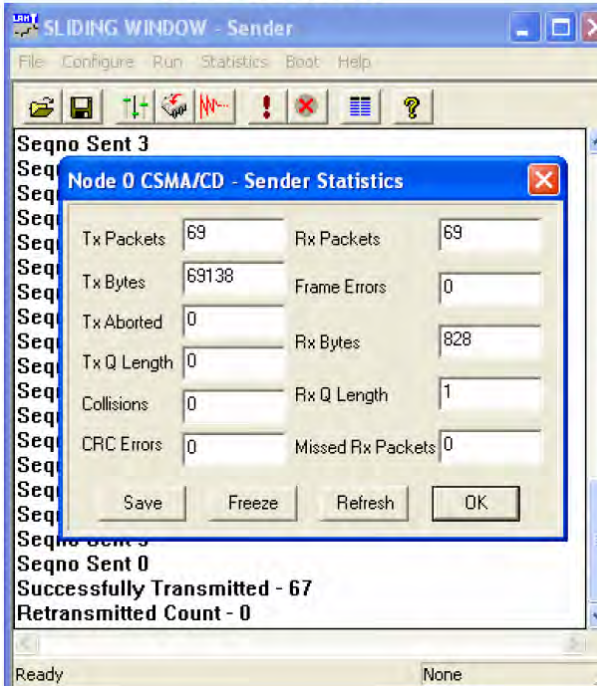
3. Set the Inter Packet Delay to 400msecs

4. Click OK button and Download the driver to the NIU using the BOOT  button command. Booting from any one of the applications is enough.

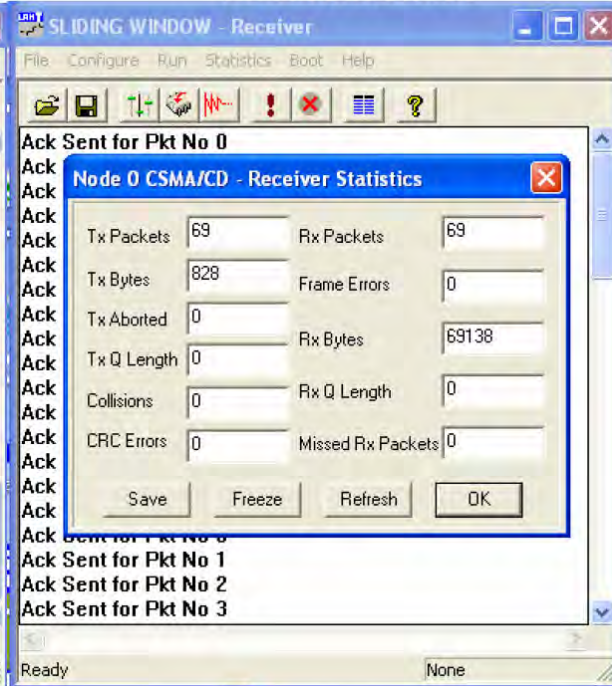
5. Run the experiment by clicking button  or by choosing RUN \_ Start from each application.

6. Set the Timeout Value to 1500 ms

**PC 1 SENDER**



**PC 2 RECEIVER**



7. Note down the no of successfully Transmitted Packets.



8. Repeat the above steps for various time out values and plot the graph between timeout Value & Throughput. Find the optimum timeout value from the plot.
9. Explain why the throughput is less compared to CSMA/CD protocol.

Calculation of Practical Throughput:

$$X = \frac{(\text{Sum of Successfully Tx packets} * \text{Packet Length} * 8)}{(\text{Duration of Experiment} * \text{Data rate})}$$

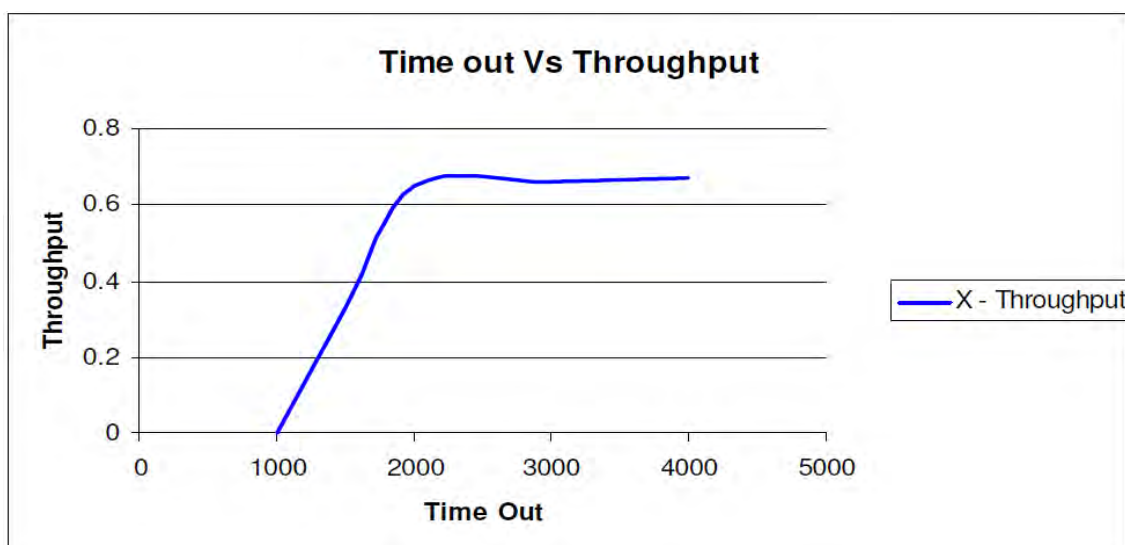
**Model Tabulation:**

**Go-Back N Protocol**

Time out value in ms	Successfully Tx packets	Practical Throughput
1000	0	0
1500	33	0.33
2000	65	0.65
3000	66	0.66
4000	67	0.67

**Model Graph:**

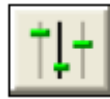
**Go-Back N Protocol**



## 5.6.2 Selective Repeat Protocol



1. Click on the Selective Repeat icon from the desktop on both PCs.



2. Click the Configuration button in the window in both the Pc's.

### PC 1 SENDER

### PC 2 RECEIVER

**Configuration View**

Node Id: 0    Duration: 100 s

Protocol: CSMA/CD    Packet Length: 1000 bytes

Baud Rate: 8K    Inter Packet Delay: 400 ms

Base Address: 0x320    No of Packets: 4

No of Nodes: 4    MyAddress: 0

Rx Mode: NON\_PROMISCUOUS MODE

I/O Mode: BLOCKING TRANSMIT

Token Release Mode: IMMEDIATE TOKEN RELEASE

Direction: Sender

Boot File Name: C:\Lantrain\Bin\Lantv13.exe

OK    Cancel

**Configuration View**

Node Id: 0    Duration: 100 s

Protocol: CSMA/CD    Packet Length: 1000 bytes

Baud Rate: 8K    Inter Packet Delay: 400 ms

Base Address: 0x320    No of Packets: 4

No of Nodes: 4    MyAddress: 0

Rx Mode: NON\_PROMISCUOUS MODE

I/O Mode: BLOCKING TRANSMIT

Token Release Mode: IMMEDIATE TOKEN RELEASE

Direction: Receiver

Boot File Name: C:\Lantrain\Bin\Lantv13.exe


OK    Cancel

Setting the configuration menu:

PC 1		PC 2	
<b>Node id</b>	0	<b>Node id</b>	0
<b>Protocol</b>	ALOHA	<b>Protocol</b>	ALOHA
<b>Baud Rate</b>	8Kbps (At both the config menu and NEU)	<b>Baud Rate</b>	8Kbps (At both the config menu and NEU)
<b>Duration</b>	100s	<b>Duration</b>	100s
<b>Packet Length</b>	1000 bytes	<b>Packet</b>	1000 bytes
<b>Bit Delay</b>	0(at NEU)	<b>Bit Delay</b>	0(at NEU)
<b>Direction</b>	Sender	<b>Direction</b>	Receiver
<b>No of packets</b>	4	<b>No of</b>	4

Note: The No of Packets parameter defines the window size.

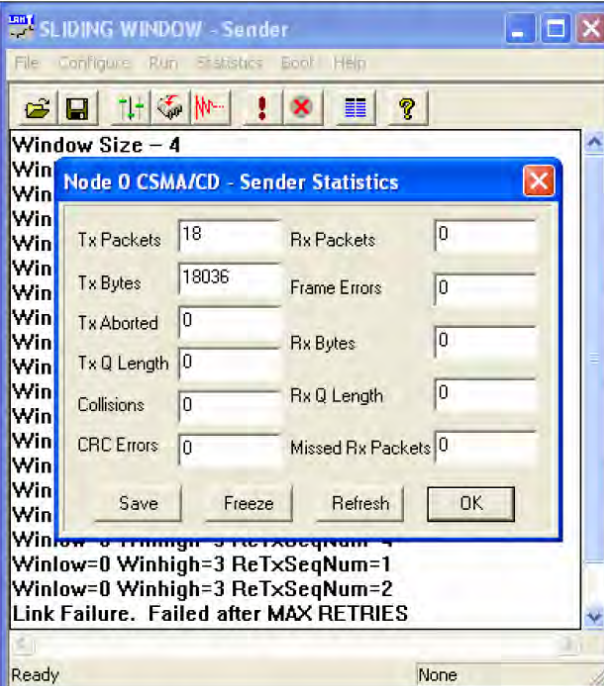
3. Set the Inter Packet Delay to 400msecs

4. Click OK button and Download the driver to the NIU using the BOOT  button command. Booting from any one of the applications is enough.

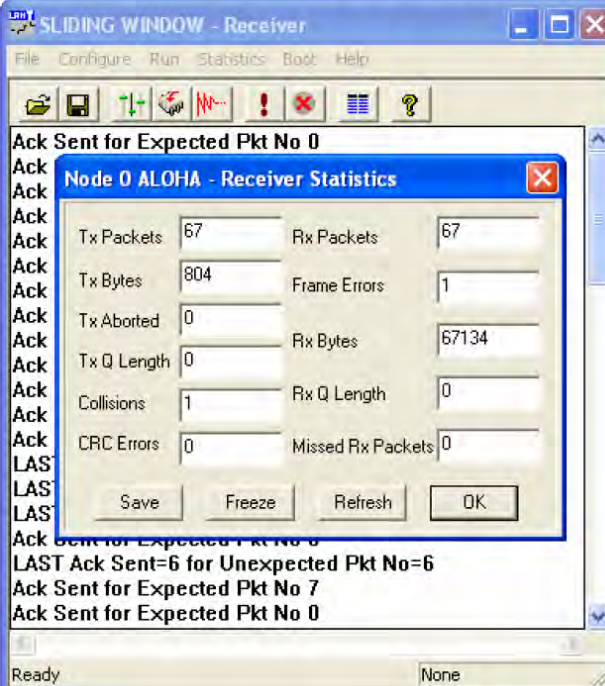
5. Run the experiment by clicking button  or by choosing RUN \_ Start from each application.

6. Set the Timeout Value to 1000 ms

**PC 1 SENDER**



**PC 2 RECEIVER**



7. Note down the no of successfully Transmitted Packets.

8. Repeat the above steps for various time out values and plot a graph between timeout Value & Throughput. Find the optimum timeout value from the plot.

Calculation of Practical Throughput:

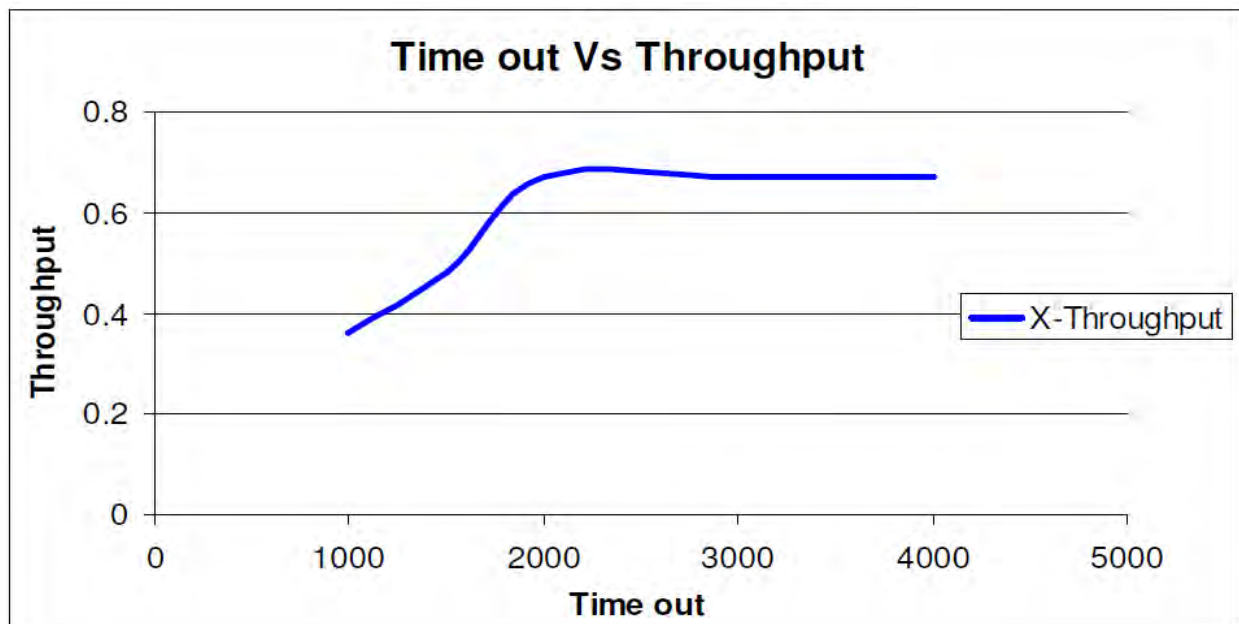
$$X = \frac{(\text{Sum of Successfully Tx packets} * \text{Packet Length} * 8)}{(\text{Duration of Experiment} * \text{Data rate})}$$

Model Tabulation:

Time out value in ms	Successfully Tx packets	Practical Throughput
1000	36	0.36
1500	48	0.48
2000	67	0.67
3000	67	0.67
4000	67	0.67

Model Graph:

Selective Repeat Protocol

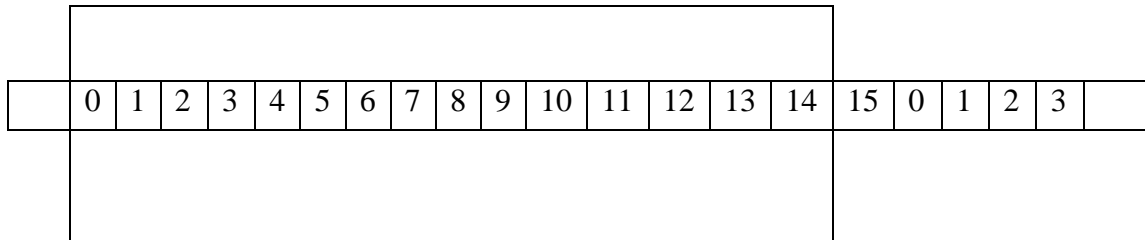


### 5.7 Post lab questions

- Draw the sender and receiver windows for a system using Go-back-N ARQ given the following .
  - Frame 0 is sent, frame 0 is acknowledged.
  - Frames 1 and 2 are sent, frames 1 and 2 are acknowledged.
  - Frame 3,4 and 5 are sent, NAK4 is received.
  - Frame 4,5,6 and 7 are sent; frames 4 through 7 are acknowledged.
- A sliding window protocol uses a sliding window of size 15. How many bits are needed to define the sequence numbers?
- Draw the sender and receiver windows for a system using selective repeat ARQ given the following.

- a. Frame 0 is sent, frame 0 is acknowledged.
- b. Frames 1 and 2 are sent, frames 1 and 2 are acknowledged.
- c. Frame 3,4 and 5 are sent, NAK4 is received.
- d. Frame 4, 5,6 and 7 are sent; frames4 through 7 are acknowledged.

4.



- a. Show the window after the sender has sent packets 0 to 11 and has received ACK8.
- b. The receiver has sent ACK6 and ACK9 but ACK6 is lost. Show the sender window.

## Lab 6: Implementation of Distance Vector Routing algorithm

**6.1 Introduction** To simulate the distance vector routing protocol to maintain routing tables as the traffic and topology of the network changes

### 6.2 Hardware Requirement

- 3PCs with NIU card
- Network Emulation Unit
- Jumper Cables

### 6.3 Background

The name distance vector is derived from the fact that routes are advertised as vectors of (distance, direction), where distance is defined in terms of a metric and direction is defined in terms of the next-hop router. For example, "Destination A is a distance of 5 hops away, in the direction of next-hop router X." As that statement implies, each router learns routes from its neighboring routers' perspectives and then advertises the routes from its own perspective. Because each router depends on its neighbors for information, which the neighbors in turn may have learned from their neighbors, and so on, distance vector routing is sometimes facetiously referred to as "routing by rumor."

The common Characteristics are

#### *Periodic Updates*

*Periodic updates* means that at the end of a certain time period, updates will be transmitted.

#### *Neighbors*

In the context of routers, *neighbors* always mean routers sharing a common data link.

#### *Broadcast Updates*

When a router first becomes active on a network, how does it find other routers and how does it announce its own presence? Several methods are available.



## Full Routing Table Updates

Most distance vector routing protocols take the very simple approach of telling their neighbors everything they know by broadcasting their entire route table, with some exceptions that are covered in following sections.

### Split Horizon

A route pointing back to the router from which packets were received is called a *reverse route*. *Split horizon* is a technique for preventing reverse routes between two routers.

## 6.4 Prelab Questions

1. What is Routing?
2. Describe about the Design Goals of Routing
3. What is a distance vector routing protocol?
4. List the common characteristics of DVR




## 6.5 Design

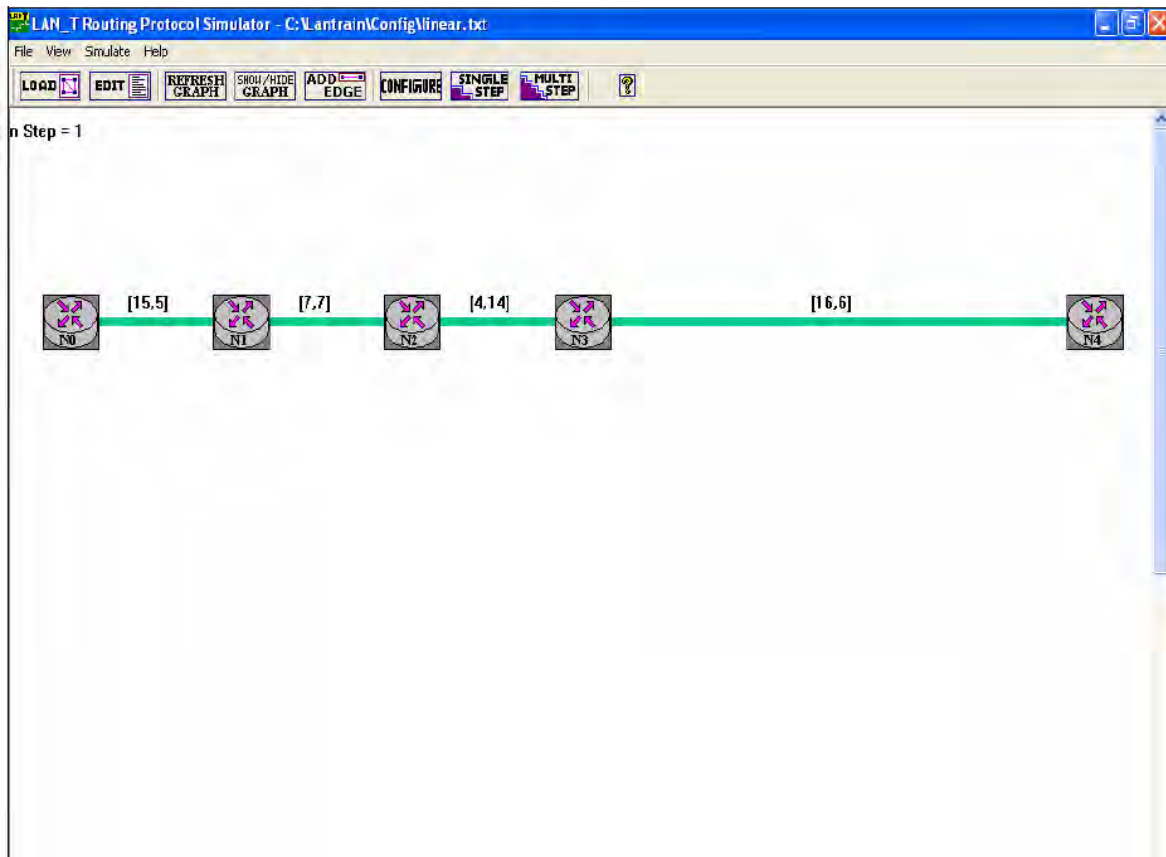
Design a network with 5 nodes in linear topology with the cost metrics for each of the link given as in graph adjacency matrix to illustrate Distance Vector Routing protocol.

### Graph adjacency matrix

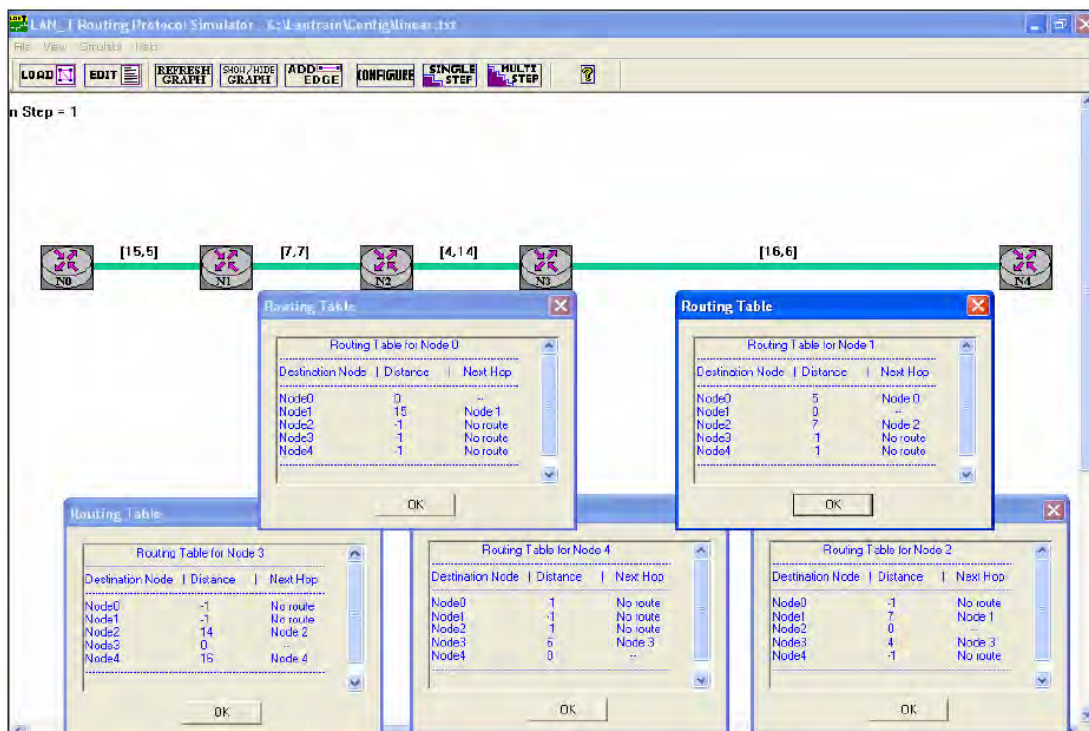
	n0	n1	n2	n3	n4
n0	0	15	-1	-1	-1
n1	5	0	7	-1	-1
n2	-1	7	0	4	-1
n3	-1	-1	14	0	16
n4	-1	-1	-1	6	0

## 6.6 Procedure:

1. Double click on LanT  Routing Simulator icon from the desktop.
2. Click button  and browse open C:\Lantrain\Config\ linear.txt.
3. Click button  and select Distance vector algorithm
4. The icon in the screen represents the nodes and the green colour line represents the path. The values inside the braces represents the 'Forward and Reverse' weights.




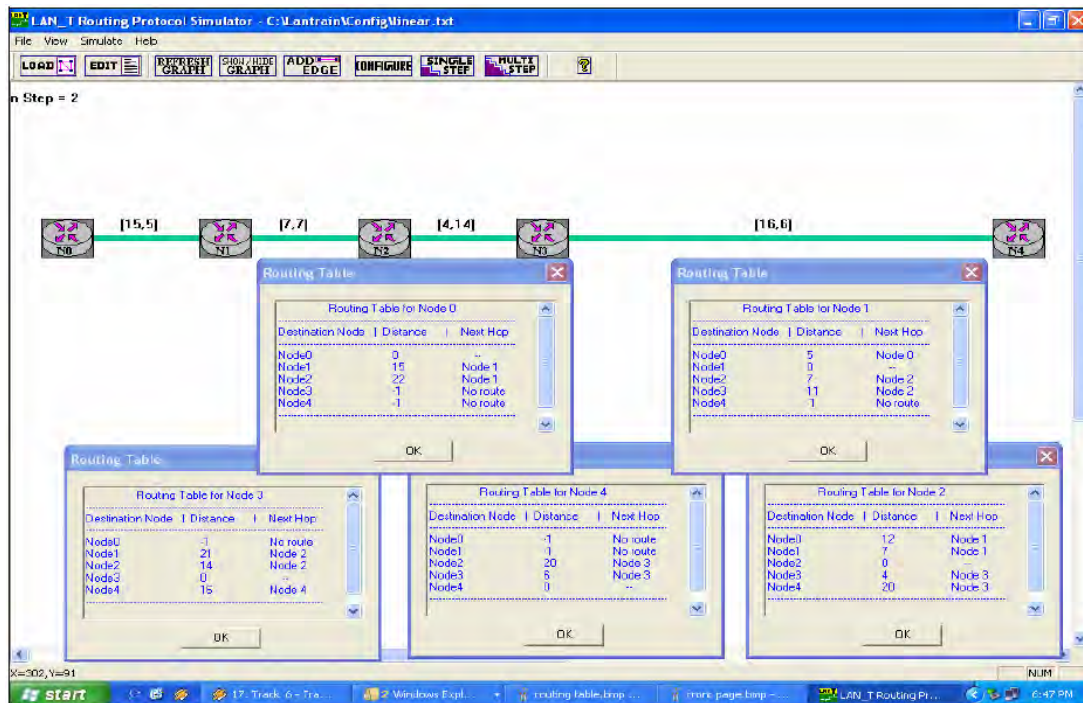
5. Click on the node icon to obtain the routing table.




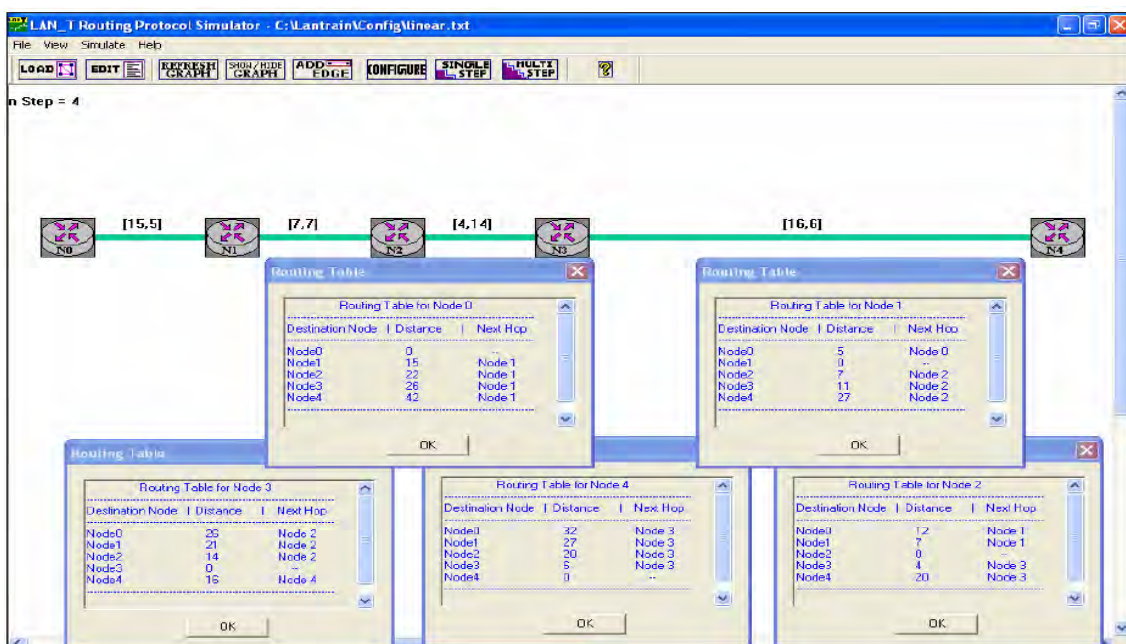
6. The above picture shows the nodes and its routing table.



7. Observe the routing table showing No route to some of the destinations even though there is a physical connection. This is because the routing table of the corresponding nodes is not been updated since there is no hopping. To update the routing table click button .



8. Hopping happens by clicking  button.

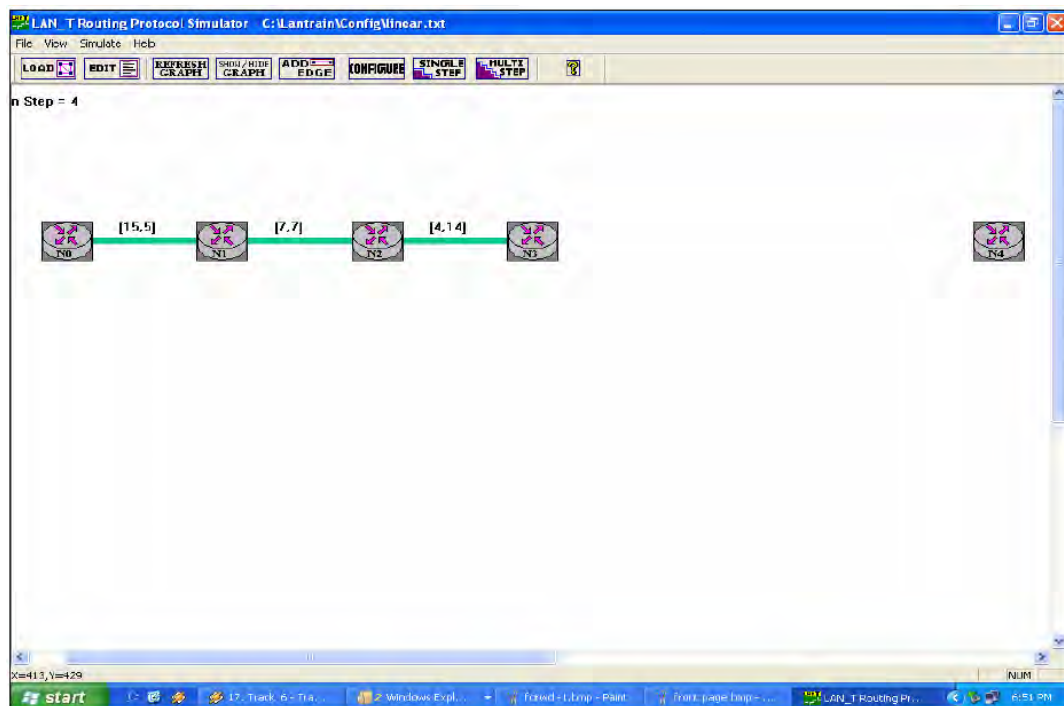
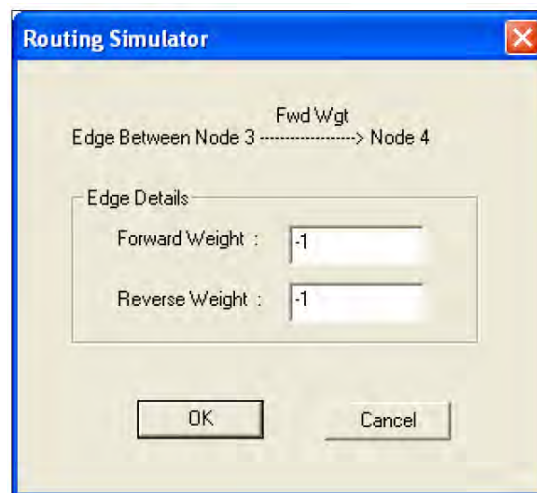


9. Now after several hopping the routing table gets updated. As the number nodes increases, the number of hopping increases. This is one of the disadvantages of distance vector algorithm.

### Count to Infinity problem

10. Click the green colour line lying between N3 and N4.

11. Enter the forward and reverse weight as '-1' in order to disconnect N4 from the other nodes



12. Now observe the routing table.

Step = 5

Routing Table for Node 0:

Destination Node	Distance	Next Hop
Node0	0	--
Node1	15	Node 1
Node2	22	Node 1
Node3	26	Node 1
Node4	42	Node 1

Routing Table for Node 1:

Destination Node	Distance	Next Hop
Node0	5	Node 0
Node1	0	--
Node2	7	Node 2
Node3	11	Node 2
Node4	27	Node 2

Routing Table for Node 2:

Destination Node	Distance	Next Hop
Node0	12	Node 1
Node1	7	Node 1
Node2	0	--
Node3	4	Node 3
Node4	20	Node 3


Routing Table for Node 3:

Destination Node	Distance	Next Hop
Node0	26	Node 2
Node1	21	Node 2
Node2	14	Node 2
Node3	0	--
Node4	16	Node 4

Routing Table for Node 4:

Destination Node	Distance	Next Hop
Node0	32	Node 3
Node1	27	Node 3
Node2	20	Node 3
Node3	6	Node 3
Node4	0	--

13. Now you could observe that there are no changes in the routing table, as they are not

updated. Click  button to update the routing table.

Step = 25

Routing Table for Node 0:

Destination Node	Distance	Next Hop
Node0	0	--
Node1	15	Node 1
Node2	22	Node 1
Node3	26	Node 1
Node4	160	Node 1

Routing Table for Node 1:

Destination Node	Distance	Next Hop
Node0	5	Node 0
Node1	0	--
Node2	7	Node 2
Node3	11	Node 2
Node4	153	Node 2

Routing Table for Node 2:

Destination Node	Distance	Next Hop
Node0	12	Node 1
Node1	7	Node 1
Node2	0	--
Node3	4	Node 3
Node4	160	Node 1

Routing Table for Node 3:

Destination Node	Distance	Next Hop
Node0	26	Node 2
Node1	21	Node 2
Node2	14	Node 2
Node3	0	--
Node4	160	Node 2

Routing Table for Node 4:

Destination Node	Distance	Next Hop
Node0	-1	No route
Node1	-1	No route
Node2	-1	No route
Node3	-1	No route
Node4	0	--

14. Even after several hopping the routing tables of N0, N1, N2, N3 shows the path and weight to N4. These false updates are another disadvantage in the 'Distance vector algorithm'.

### **6.7 Post Lab Questions**

1. Name several problems associated with distance vector protocols
2. What is counting-to-infinity problem, and how can it be controlled?
3. Describe about the various Routing Metrics

## Lab 7 – Link State routing protocol

### 7.1 Introduction:

To simulate the link state routing protocol to maintain routing tables as the traffic and topology of the network changes

### 7.2 Hardware Requirement

- 3PCs with NIU card
- Network Emulation Unit
- Jumper Cables

### 7.3 Background

The information available to a distance vector router has been compared to the information available from a road sign. Link state routing protocols are like a road map. A link state router cannot be fooled as easily into making bad routing decisions, because it has a complete picture of the network. The reason is that unlike the routing-by-rumor approach of distance vector, link state routers have firsthand information from all their peer<sup>7</sup> routers. Each router originates information about itself, its directly connected links, and the state of those links (hence the name). This information is passed around from router to router, each router making a copy of it, but never changing it. The ultimate objective is that every router has identical information about the internetwork, and each router will independently calculate its own best paths. Link state protocols, sometimes called *shortest path first* or *distributed database* protocols, are built around a well-known algorithm from graph theory, E. W. Dijkstra's shortest path algorithm. Examples of link state routing protocols are:

- Open Shortest Path First (OSPF) for IP
- The ISO's Intermediate System to Intermediate System (IS-IS) for CLNS and IP
- DEC's DNA Phase V
- Novell's NetWare Link Services Protocol (NLSP)

Although link state protocols are rightly considered more complex than distance vector protocols, the basic functionality is not complex at all:



1. Each router establishes a relationship—an adjacency—with each of its neighbors.
2. Each router sends *link state advertisements* (LSAs), some
3. Each router stores a copy of all the LSAs it has seen in a database. If all works well, the databases in all routers should be identical.
4. The completed *topological database*, also called the *link state database*, describes a graph of the internetwork. Using the Dijkstra algorithm, each router calculates the shortest path to each network and enters this information into the route table.

#### 7.4 Prelab Questions

1. Explain the procedure to calculate Shortest path in LSR protocol
2. What is the purpose of a topological database?
3. Why are sequence numbers important in link state protocols?


#### 7.5 Design


Design a network with 5 nodes in linear topology with the cost metrics for each of the link given as in graph adjacency matrix to interpret how count to infinity problem is rectified.


#### Graph adjacency matrix

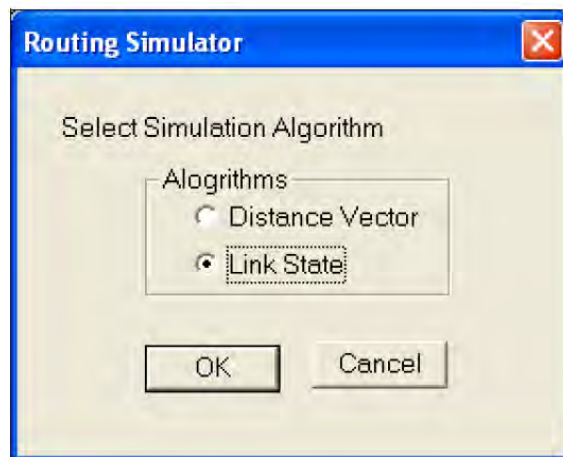
	n0	n1	n2	n3	n4
n0	0	15	-1	-1	-1
n1	5	0	7	-1	-1
n2	-1	7	0	4	-1
n3	-1	-1	14	0	16
n4	-1	-1	-1	6	0

#### 7.6 Procedure:

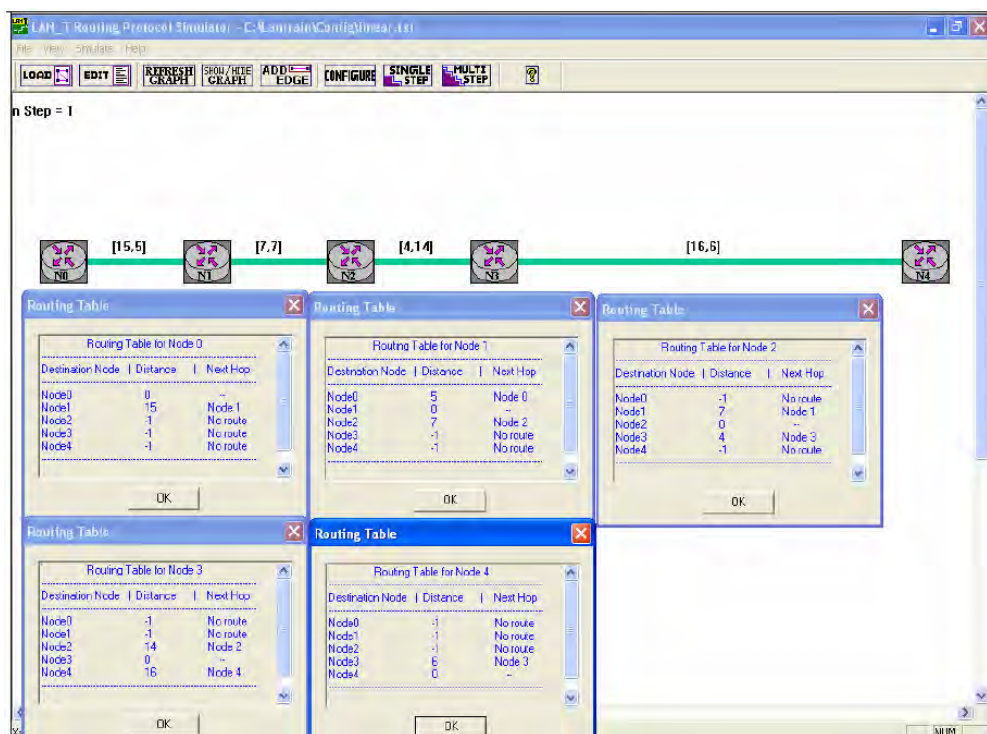
1. Double click on LanT Routing Simulator icon  from the desktop.


2. Click  button and browse open C:\Lantrain\Config\linear.txt.

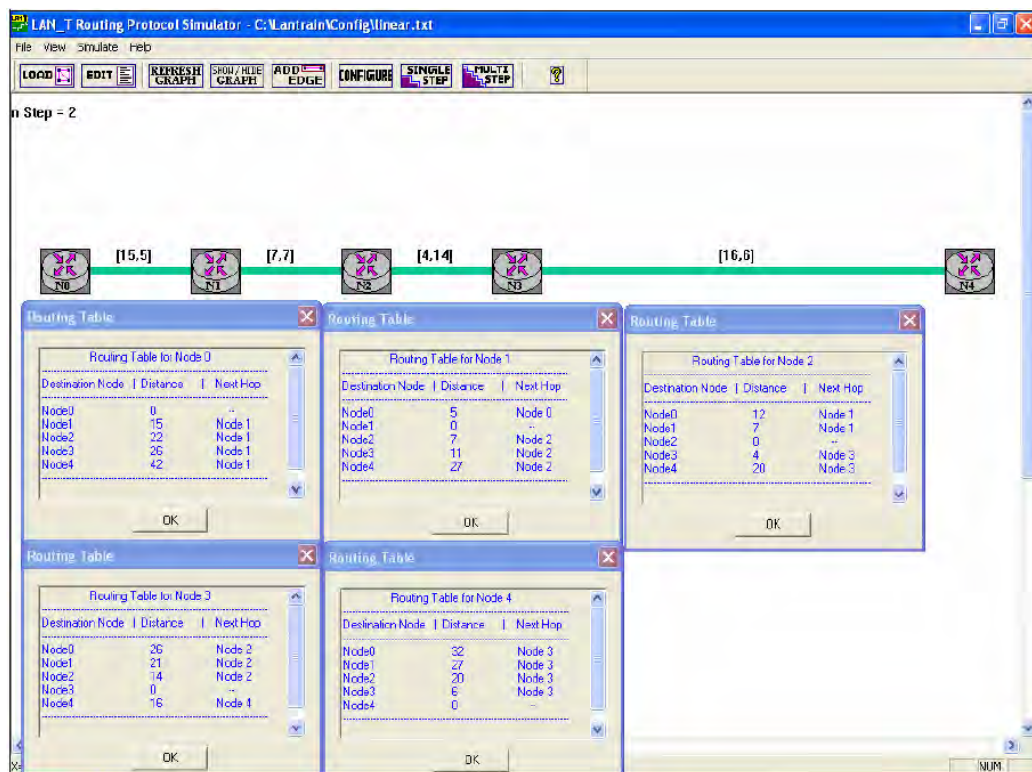
3. Click  button and select Link state algorithm



4. Click on the nodes to obtain the routing table.



5. Click the  button to update the routing table. Routing table of entire nodes gets updated after a single hopping. This is one of the advantages of Link state algorithm over Distance vector algorithm.



Count to Infinity problem:

6. Click on the green colour line lying between N3 and N4.
7. Enter forward and reverse weights as -1 to disconnect N4 from the other nodes.

The screenshot shows the 'Routing Simulator' dialog box. It displays the 'Fwd Wgt' for the edge between Node 3 and Node 4. The 'Edge Details' section contains two input fields: 'Forward Weight' and 'Reverse Weight', both of which are currently set to -1. The dialog box has 'OK' and 'Cancel' buttons at the bottom.

Routing Simulator

Fwd Wgt

Edge Between Node 3 -----> Node 4

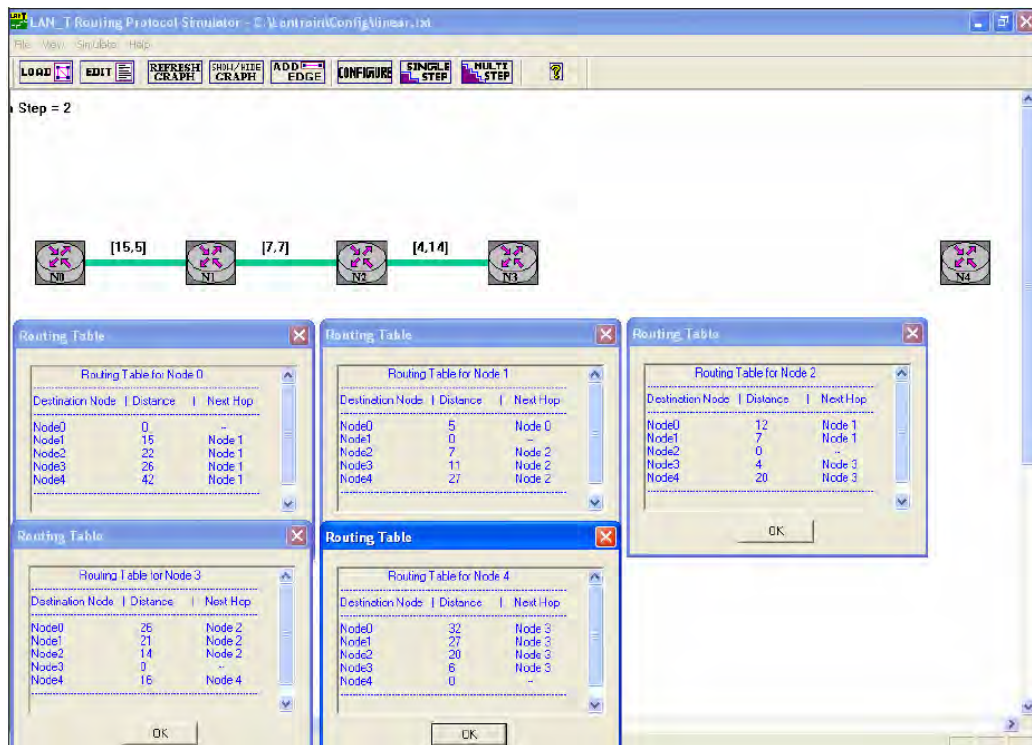
Edge Details

Forward Weight : -1

Reverse Weight : -1

OK Cancel

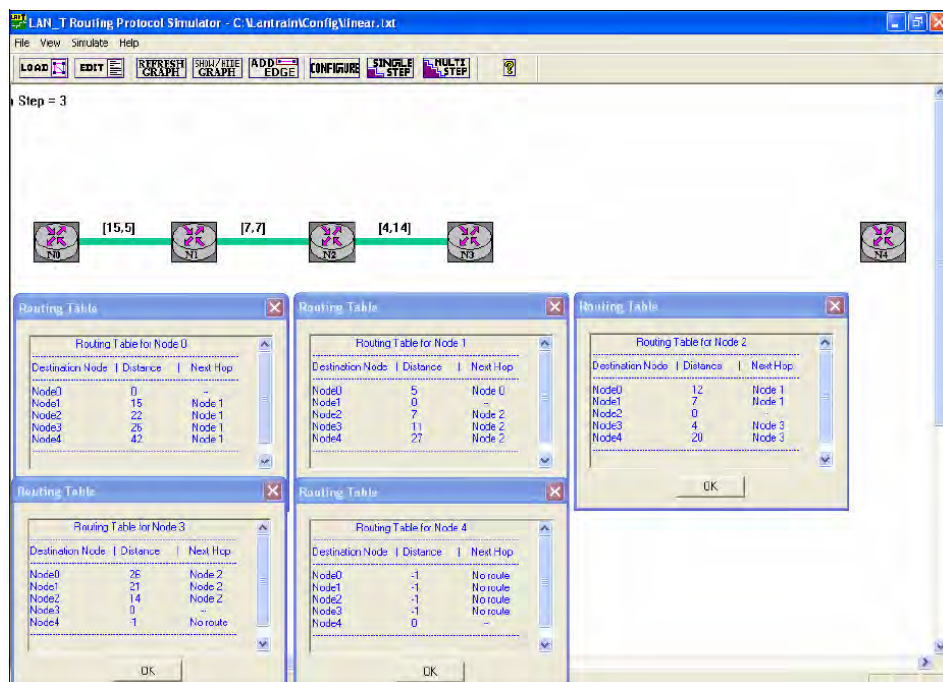




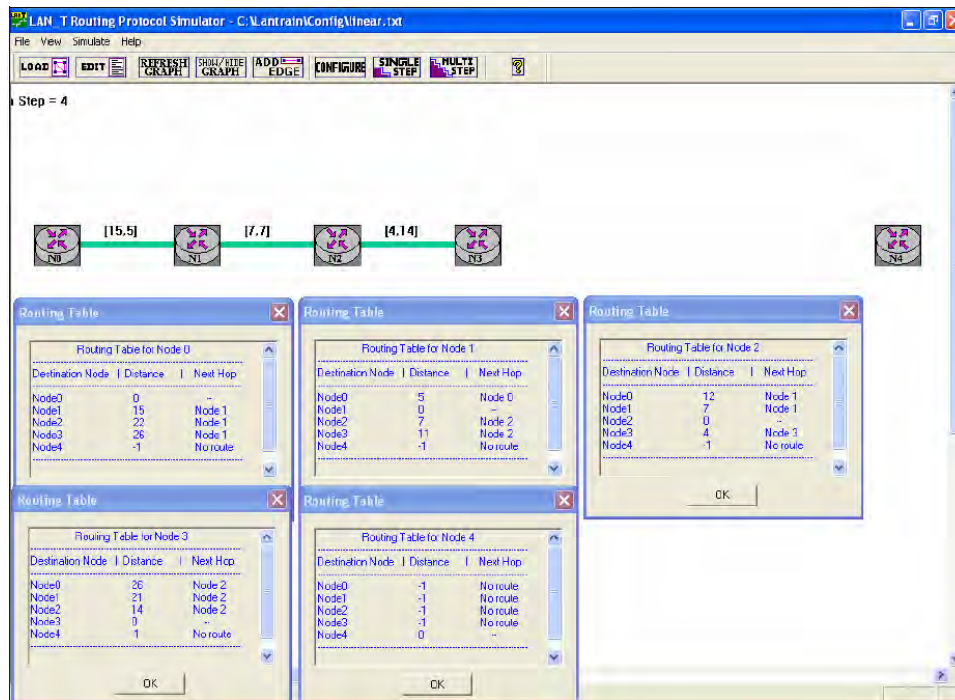
8. Observe the routing table. The values are not changed as it's not updated.



9. Click the single step button.



10. Now you could see the routing table for each nodes been updated. This is the advantage of 'Link state algorithm' over 'Distance vector algorithm'.



## 7.7 Post Lab Questions

1. What are the differences between distance vector and link state routing protocols?
2. Explain the Steps involved in building a Routing Table using Dijkstra's Algorithm
3. Can topology be built using partial knowledge of the each node? If yes explain why?

## Lab 8: Encrypting a file using RC4 algorithm

### 8.1 Introduction:

The purpose of this experiment is to introduce you to RC4 algorithm, encryption and decryption.

### 8.2 Hardware Requirement

- 3PCs with NIU card
- Network Emulation Unit
- Jumper Cables

### 8.3 Background

In cryptography, RC4 (also known as ARC4 or ARCFOUR meaning Alleged RC4) is the most widely-used software stream cipher and is used in popular protocols such as Secure Sockets Layer (SSL) (to protect Internet traffic) and WEP (to secure wireless networks). It is remarkable for its simplicity and speed in software. RC4 generates a pseudorandom stream of bits (a keystream) which, for encryption, is combined with the plaintext using bit-wise exclusive-or; decryption is performed the same way (since exclusive-or is a symmetric operation). (This is similar to the Vernam cipher except that generated *pseudorandom bits*, rather than a prepared stream, are used.) To generate the keystream, the cipher makes use of a secret internal state which consists of two parts:

1. A permutation of all 256 possible bytes (denoted "S" below).
2. Two 8-bit index-pointers (denoted "i" and "j").

The permutation is initialized with a variable length key, typically between 40 and 256 bits, using the *key-scheduling* algorithm (KSA). Once this has been completed, the stream of bits is generated using the *pseudo-random generation algorithm* (PRGA).

### 8.4 Pre lab questions

1. What does RC4 stands for?
2. What is Key scheduling algorithm and Pseudo Random Generation Algorithm?
3. What is Cipher? Explain the steps in Generation of Ciphers

### 8.5 Design

Encrypt the word HELLO using RC4 Encryption Algorithm.

## 8.6 Procedure

### Initial set-up:

Install jre 1.5(java run time environment) to run this application.

Understand the Steps Involved in RC4 Encryption and Decryption Algorithm.



1. Double click on RC4Key icon.

**STREAM CIPHER - RC4 ALGORITHM**

**Mode**  
☐ Step  
☒ Full

Text - T

**Key - K**  
☒  (Text)  
☐  (Binary)

SBox

KBox

SBox Permutation

Steps

Next Clear

**Key Generation Phase**  
SBox[i] & SBox[j]   
Key BitStream   
Binary Value (T)   
XOR value   
Cipher text

2. Select Step mode and type any text to encrypt. Type any encryption key in either text or binary mode.
3. Click the Next button from the right side panel.

**STREAM CIPHER - RC4 ALGORITHM**

**Mode**

☒ Step

☐ Full

Text - T

**Key - K**

☒ 1234 (Text)

☐ (Binary)

SBox Permutation

**SBox**

196	197	198	199	200	201	202	203	204
205	206	207	208	209	210	211	212	213
214	215	216	217	218	219	220	221	222
223	224	225	226	227	228	229	230	231
232	233	234	235	236	237	238	239	240
241	242	243	244	245	246	247	248	249
250	251	252	253	254	255			

**KBox**

50	51	52	49	50	51	52	49	50	51	52	49	50
51	52	49	50	51	52	49	50	51	52	49	50	51
52	49	50	51	52	49	50	51	52	49	50	51	52
49	50	51	52	49	50	51	52	49	50	51	52	49
50	51	52	49	50	51	52	49	50	51	52	49	50
51	52	49	50	51	52	49	50	51	52	49	50	51
52	49	50	51	52	49	50	51	52	49	50	51	52

**Steps**

Step 1.  
Plain Text's Binary value is displ.

Step 2.  
Initialization of State Table (SBox)

Step 3.  
Initialization of Key Table (KBox)

**Key Generation Phase**

SBox[j] & SBox[j]

Key BitStream

Binary Value (T) 01101000 01100101 01101100 01101100 01101111

XOR value

Cipher text

**STREAM CIPHER - RC4 ALGORITHM**

**Mode**

☒ Step

☐ Full

Text - T

**Key - K**

☒ 1234 (Text)

☐ (Binary)

SBox Permutation

**SBox**

196	197	198	199	200	201	202	203	204
205	206	207	208	209	210	211	212	213
214	215	216	217	218	219	220	221	222
223	224	225	226	227	228	229	230	231
232	233	234	235	236	237	238	239	240
241	242	243	244	245	246	247	248	249
250	251	252	253	254	255			

**KBox**

50	51	52	49	50	51	52	49	50	51	52	49	50
51	52	49	50	51	52	49	50	51	52	49	50	51
52	49	50	51	52	49	50	51	52	49	50	51	52
49	50	51	52	49	50	51	52	49	50	51	52	49
50	51	52	49	50	51	52	49	50	51	52	49	50
51	52	49	50	51	52	49	50	51	52	49	50	51
52	49	50	51	52	49	50	51	52	49	50	51	52

**Steps**

Step 1.  
Plain Text's Binary value is displ.

Step 2.  
Initialization of State Table (SBox)

Step 3.  
Initialization of Key Table (KBox)

**Key Generation Phase**

SBox[j] & SBox[j]

Key BitStream

Binary Value (T) 01101000 01100101 01101100 01101100 01101111

XOR value

Cipher text

4. Now the State table and Key tables are formed. (S Box contains a matrix of 256 values starting from 0 and K Box contains the matrix of repeated ASCII values of the key entered).
5. The flow of the algorithm in each step is explained in the description panel on the right side.
6. In the next step the permuted values are stored in the S Box Permutation.
7. Key bit stream is generated from the values obtained from S Box Permutation.



### STREAM CIPHER - RC4 ALGORITHM

**Mode**

☒ Step

☐ Full

Text - T

**Key - K**

☒ 1234 (Text)

(Binary)

**SBox**

196	197	198	199	200	201	202	203	204
205	206	207	208	209	210	211	212	213
214	215	216	217	218	219	220	221	222
223	224	225	226	227	228	229	230	231
232	233	234	235	236	237	238	239	240
241	242	243	244	245	246	247	248	249
250	251	252	253	254	255			

**KBox**

50	51	52	49	50	51	52	49	50	51	52	49	50
51	52	49	50	51	52	49	50	51	52	49	50	51
52	49	50	51	52	49	50	51	52	49	50	51	52
49	50	51	52	49	50	51	52	49	50	51	52	49
50	51	52	49	50	51	52	49	50	51	52	49	50
51	52	49	50	51	52	49	50	51	52	49	50	51
52	49	50	51	52	49	50	51	52	49	50	51	52

**SBox Permutation**

195	26	161	209	61	62	151	153	217	3	86	94
164	47	183	176	249	2	144	148	28	227	206	
107	194	9	177	20	205	27	6	242	10	90	241
89	40	140	125	64	240	208	228	22	231	139	
181	213	52	198	141	76	46	42	204	92	193	
137	120	43	170	180	156	152	48	220	219	191	
54	85	70	60								

**Steps**

Initial Permutation of SBox as

```

j = 0;
for i = 0 to 255 do
  j = ( j + S[i] + K[i] ) mod 256
  swap ( S[i], S[j] )

```

Step 5.

Key Stream Generation

```

i,j=0;
while (true)
  i = (i+1) mod 256;
  j = ( j + S[i] ) mod 256;
  swap ( S[i], S[j] );
  t = (S[i] + S[j]) mod 256;
  k = S[t];

```

**Next** **Clear**

**Key Generation Phase**

SBox[i] & SBox[j] 100 160 22 67 164 225 2 5 2 59

Key BitStream 00000101 01001001 01110001 10101111 11110011

Binary Value (T) 01101000 01100101 01101100 01101100 01101111

XOR value

Cipher text

8. XOR operation is done between the binary value and key bit stream generated..

### STREAM CIPHER - RC4 ALGORITHM

**Mode**

☒ Step

☐ Full

Text - T

**Key - K**

☒ 1234 (Text)

(Binary)

**SBox**

196	197	198	199	200	201	202	203	204
205	206	207	208	209	210	211	212	213
214	215	216	217	218	219	220	221	222
223	224	225	226	227	228	229	230	231
232	233	234	235	236	237	238	239	240
241	242	243	244	245	246	247	248	249
250	251	252	253	254	255			

**KBox**

50	51	52	49	50	51	52	49	50	51	52	49	50
51	52	49	50	51	52	49	50	51	52	49	50	51
52	49	50	51	52	49	50	51	52	49	50	51	52
49	50	51	52	49	50	51	52	49	50	51	52	49
50	51	52	49	50	51	52	49	50	51	52	49	50
51	52	49	50	51	52	49	50	51	52	49	50	51
52	49	50	51	52	49	50	51	52	49	50	51	52

**SBox Permutation**

195	26	161	209	61	62	151	153	217	3	86	94
164	47	183	176	249	2	144	148	28	227	206	
107	194	9	177	20	205	27	6	242	10	90	241
89	40	140	125	64	240	208	228	22	231	139	
181	213	52	198	141	76	46	42	204	92	193	
137	120	43	170	180	156	152	48	220	219	191	
54	85	70	60								

**Steps**

```

j = ( j + S[i] + K[i] ) mod 256
swap ( S[i], S[j] )

```

Step 5.

Key Stream Generation

```

i,j=0;
while (true)
  i = (i+1) mod 256;
  j = ( j + S[i] ) mod 256;
  swap ( S[i], S[j] );
  t = (S[i] + S[j]) mod 256;
  k = S[t];

```

Step 6.

XOR value obtained ( keystream )

**Next** **Clear**

**Key Generation Phase**

SBox[i] & SBox[j] 100 160 22 67 164 225 2 5 2 59

Key BitStream 00000101 01001001 01110001 10101111 11110011

Binary Value (T) 01101000 01100101 01101100 01101100 01101111

XOR value 01101101 00101100 00011101 11000011 10011100

Cipher text

9. Cipher text is formed corresponding to the XOR value generated.

**STREAM CIPHER - RC4 ALGORITHM**

**Mode**  
☒ Step  
☐ Full

**Text - T**

**Key - K**  
☒ 1234 (Text)  
☐ (Binary)

**SBox**  
196 197 198 199 200 201 202 203 204  
205 206 207 208 209 210 211 212 213  
214 215 216 217 218 219 220 221 222  
223 224 225 226 227 228 229 230 231  
232 233 234 235 236 237 238 239 240  
241 242 243 244 245 246 247 248 249  
250 251 252 253 254 255

**KBox**  
50 51 52 49 50 51 52 49 50 51 52 49 50  
51 52 49 50 51 52 49 50 51 52 49 50 51  
52 49 50 51 52 49 50 51 52 49 50 51 52  
49 50 51 52 49 50 51 52 49 50 51 52 49  
50 51 52 49 50 51 52 49 50 51 52 49 50  
51 52 49 50 51 52 49 50 51 52 49 50 51  
52 49 50 51 52 49 50 51 52

**SBox Permutation**  
195 26 161 209 61 62 151 153 217 3 86 94  
164 47 183 176 249 2 144 148 28 227 206  
107 194 9 177 20 205 27 6 242 10 90 241  
89 40 140 125 64 240 208 228 22 231 139  
181 213 52 198 141 76 46 42 204 92 193  
137 120 43 170 180 156 152 48 220 219 191  
54 85 70 60

**Steps**  
Step 3.  
Key Stream Generation  
i,j=0;  
while (true)  
i = (i+1) mod 256;  
j = (j + S[i]) mod 256;  
swap ( S[i], S[j]);  
t = (S[i] + S[j]) mod 256;  
k = S[t];  
  
Step 6.  
XOR value obtained ( keystream )  
  
Step 7.  
character representation of cipher text

**Key Generation Phase**  

SBox[i] & SBox[j]	100 160	22 67	164 225	2 5	2 59
Key BitStream	00000101 01001001 01110001 10101111 11110011				
Binary Value (T)	01101000 01100101 01101100 01101100 01101111				
XOR value	01101101 00101100 00011101 11000011 10011100				
Cipher text	m, □ Å □				

10. Copy the cipher text by selecting and pressing ctrl + c.
11. Click the clear button and paste the cipher text in the text field.
12. Enter the same key that is used for encryption.

**STREAM CIPHER - RC4 ALGORITHM**

**Mode**  
☒ Step  
☐ Full

**Text - T**

**Key - K**  
☒ 1234 (Text)  
☐ (Binary)

**SBox**  
196 197 198 199 200 201 202 203 204  
205 206 207 208 209 210 211 212 213  
214 215 216 217 218 219 220 221 222  
223 224 225 226 227 228 229 230 231  
232 233 234 235 236 237 238 239 240  
241 242 243 244 245 246 247 248 249  
250 251 252 253 254 255

**KBox**  
50 51 52 49 50 51 52 49 50 51 52 49 50  
51 52 49 50 51 52 49 50 51 52 49 50 51  
52 49 50 51 52 49 50 51 52 49 50 51 52  
49 50 51 52 49 50 51 52 49 50 51 52 49  
50 51 52 49 50 51 52 49 50 51 52 49 50  
51 52 49 50 51 52 49 50 51 52 49 50 51  
52 49 50 51 52 49 50 51 52

**SBox Permutation**  
195 26 161 209 61 62 151 153 217 3 86 94  
164 47 183 176 249 2 144 148 28 227 206  
107 194 9 177 20 205 27 6 242 10 90 241  
89 40 140 125 64 240 208 228 22 231 139  
181 213 52 198 141 76 46 42 204 92 193  
137 120 43 170 180 156 152 48 220 219 191  
54 85 70 60

**Steps**  
Step 5.  
Key Stream Generation  
i,j=0;  
while (true)  
i = (i+1) mod 256;  
j = (j + S[i]) mod 256;  
swap ( S[i], S[j]);  
t = (S[i] + S[j]) mod 256;  
k = S[t];  
  
Step 6.  
XOR value obtained ( keystream )  
  
Step 7.  
character representation of cipher text

**Key Generation Phase**  

SBox[i] & SBox[j]	100 160	22 67	164 225	2 5	2 59
Key BitStream	00000101 01001001 01110001 10101111 11110011				
Binary Value (T)	01101101 00101100 00011101 11000011 10011100				
XOR value	01101000 01100101 01101100 01101100 01101111				
Cipher text	hello				

13. Now you could retrieve back the original message.

### **8.7 Post Lab Questions**

1. Using the RC4 Encryption scheme, encrypt a character with a key and observe the cipher text. Repeat the same procedure for a different key but with the same text and see if there is any relation between them. Repeat with two same characters and observe whether the cipher text also gets repeated.
2. Explain in Detail the generation of key in Key Scheduling Algorithm



## **Lab 9: Transfer of files from PC to PC using Windows/ UNIX socket processing**

### **9.1 Introduction**

The purpose of this lab is to introduce you the concept of socket programming, TCP connection establishment, and Session management. In this lab you will be able to establish a socket connection between two computers and use it to reliably transfer a file.

### **9.2 Hardware Requirement**

Two computers with Net Sim software

### **9.3 Background**

File transfer uses the client –server paradigm. The server waits passively listening for requests from a client. After servicing the request, the server resumes listening. The client is usually interactive and accepts commands specifying the file name, whether to or from the server, username and password etc.

File transfer is a session oriented task. It has three phases:

1. Session establishment: Setup a logical connection between client and server.
2. Data transfer: the file is transferred to /from the server, with appropriate error checking.
3. Session termination: the files are closed and the logical connection taken down.

Three phases are described in detail below.

#### **Session Establishment**

This involves several steps.

1. Setup of a transport layer connection with the server. This is usually TCP.
2. Sending username and password to the server for authentication
3. Checking the file permissions on both ends. E.g to send a file F, the local file F must exist and be readable by the user, and the remote file F must be writable by the user. In case “no overwrite mode” is specified, the remote file F must not exist.

4. Negotiating session parameters such as the file type (text or binary) and block size.

### **Data Transfer**

The sender reads the file one segment at a time and sends it on the TCP connection. The segment size may be large, e.g. 10s to 100s of KB. Note that some TCP implementations may limit the size of each segment, say to 64 KB. TCP provides reliable data transfer. However, it may abruptly close the connection without delivering the data to the remote application. To handle such situations, the file transfer receiver send an ACK to the sender after successfully writing the data to the destination file.

### **Session Termination**

This may be initiated by either side. The initiator closes its local files and sends a disconnect request to the other side. The other party closes its file and responds with a disconnect confirm. Each side closes its TCP socket. The server resumes listening for connect requests.

A session may be terminated normally after successful transfer of the file or abnormally after incomplete transfer such as due to network failure. In the latter case, the receiver should delete the incomplete file so that it is not inadvertently used.

### **Error Handling**

For any operation where an ACK is expected, the sender starts a timer. In case of expiry of the timer, the operation may be retried or is aborted and an error indication returned.

## **9.4 Pre lab questions**

1. What is socket programming?
2. What are the phases involved in file transfer?
3. List the features of TCP model.
4. What is DHCP?
5. Give the IP address range of class A, B, C, D.

## 9.5 Design

Establish a socket connection between two computers of IP addresses 10.1.114.x and 10.1.114.y. Use the established connection to reliably transfer a file of size less than or equal to 1024 bytes.

## 9.6 Procedure

### Initial set-up:

Copy and paste **FTPGui** folder in **C:** and **SocketFTP.exe** file in **C:\Lantrain\bin\**

Connect the Ethernet (RJ-45) ports of two PCs with the supplied swap cable.

Since these 2 PCs are connected and there is no DHCP so to assign IP address, setup two PCs

IP address and gateway as follows

#### **1<sup>st</sup> PC:**

IP address : 192.168.0.1

Gateway : 192.168.0.1

#### **2<sup>nd</sup> PC:**


IP address : 192.168.0.2

Gateway : 192.168.0.2

Check if there is a connection between them by opening a browser and typing the other PCs IP address. The other PCs Directory structure will be displayed in the browser and this is the indication that the connection is established between the PCs.

### Procedure:



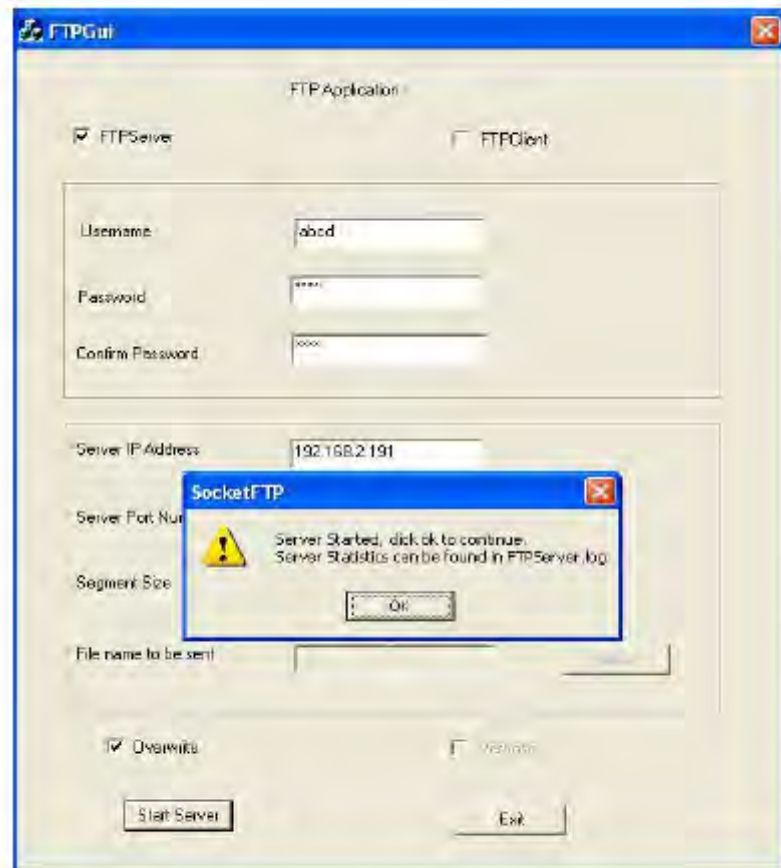
1. Double click on  icon from the desktop.
2. Setup username, password, server ip address, and server port no in the application.

The screenshot shows the FTPGui application window. At the top, it says "FTP Application". Below this, there are two radio buttons: "FTPServer" (which is selected) and "FTPClient". Underneath, there are three text input fields for "Username", "Password", and "Confirm Password", all of which are empty. Below these, there are four more text input fields: "Server IP Address", "Server Port Number" (containing "0"), "Segment Size" (containing "0"), and "File name to be sent". To the right of the "File name to be sent" field is a "Browse" button. At the bottom, there are two checkboxes: "Overwrite" (checked) and "Verbose" (unchecked). Finally, there are two buttons: "Start Server" and "Exit".

This screenshot shows the same FTPGui application window, but with the input fields filled with example data. The "Username" field contains "abod", the "Password" field contains "1234", and the "Confirm Password" field contains "1234". The "Server IP Address" field contains "192.168.2.191", the "Server Port Number" field contains "2000", and the "Segment Size" field contains "0". The "File name to be sent" field is still empty, and the "Browse" button remains. The "Overwrite" checkbox is still checked, and the "Verbose" checkbox remains unchecked. The "Start Server" and "Exit" buttons are still present at the bottom.

3. Ensure that the **FTPServer** and **Overwrite** check boxes are checked.

4. Click **Start Server** button.



5. Click ok to continue.
6. Open FTP client in another PC.



7. Check the **FTPClient** and **Verbose** check box.
8. Give the same username, password, server ip address, port number and segment size (say 128 or 256 or 1024) and specify the file that has to be sent.
9. Click **Send File**.
10. Now you could find your file in **C:\FTPGui** of the server.

### **Exercises:**

- ✓ In this exercise, you will verify the various features of the server. In each case, select a file F and proceed as directed. Note the reason for failure (if any)
- ✓ Give the correct username but wrong password.
- ✓ Give the wrong username but correct password.
- ✓ Give the correct username and password and transfer the file. Now, repeat with overwrite mode turned off.
- ✓ Using the file manager on the server, make the file F read-only. Now, on the client turn overwrite mode on and transfer the file.
- ✓ Specify the wrong server IP address and port number.

- 
- ✓ Select a file of size at least 1MB, set verbose mode on and send it to the server. Note the time taken and compute,  $\text{throughput} = \text{file size} / \text{time taken}$ . Repeat the experiment with verbose mode off. Compare the throughputs and explain the difference.
  - ✓ With verbose mode off, compute the throughput for files of size 1B, 1000 B, 10 KB, 1 MB, and 10 MB. Explain the differences. (Note: it will help to plot a graph of throughput versus file size, with a logarithmic x-axis).
  - ✓ Start a transfer of a large file with verbose mode on. Immediately unplug the Ethernet cable. After couple of seconds plug the cable back. Compute the throughput. Now repeat, except do not plug in the cable. Note the time taken for the client to time out and give up.

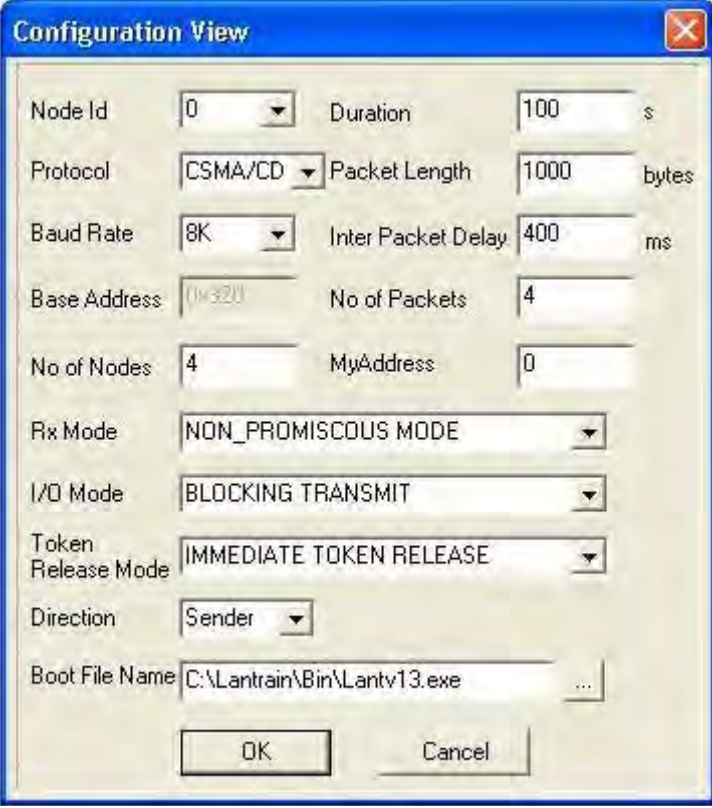
### **9.7 Post lab questions**

1. What is the difference between TCP and UDP protocol?
2. What are the timers available in TCP?
3. What do you mean by socket address?
4. Give the port numbers of TCP and UDP.
5. Given the IP address: 192.168.0.1. Identify the range and net id.



## Appendix

### Configuration menu details



The screenshot shows a 'Configuration View' dialog box with the following fields and values:

Field	Value	Unit/Label
Node Id	0	
Duration	100	s
Protocol	CSMA/CD	
Packet Length	1000	bytes
Baud Rate	8K	
Inter Packet Delay	400	ms
Base Address	(1132)	
No of Packets	4	
No of Nodes	4	
MyAddress	0	
Rx Mode	NON_PROMISCOUS MODE	
I/O Mode	BLOCKING TRANSMIT	
Token Release Mode	IMMEDIATE TOKEN RELEASE	
Direction	Sender	
Boot File Name	C:\Lantrain\Bin\Lantv13.exe	

Buttons: OK, Cancel

**Config View** displays the configuration dialog box where all the parameter values are displayed.

**Node Id** - Specify that that particular application/experiment window should be as node 0 or 1. This is the one that differentiates the two applications in the same PC.

**Protocol** - Specify the MAC Protocol that NIU has to emulate. By default, it is set to Aloha

**Baud Rate** - Specify the data rate between 8Kbps and 1Mbps. (Only for CSMA/CD and Ring Mode, this needs to be set both in the Application window and the NEU. For Aloha and CSMA, settings in NEU are sufficient.)

**No of Nodes** - Specify the number of nodes in the network that are used to do Token Ring Experiment. This field may be used for other purposes while writing codes for suggested Experiments in exercise also.

**Duration** - Specify the duration of the experiment in seconds.

**Packet Length** - Specify the length of packets that are used in the experiment. Can be set to a maximum of 1000 bytes.

**Inter Packet Delay** - Specify the inter packet delay (that is equivalent to the inter packet arrival time) based on the calculation suggested in the experiments. This has to be in milliseconds.

**Number of Packets** - Specify number of packets in a window for Sliding Window experiment. This field may be used for other purposes while writing codes for suggested experiments in exercise, say for example, to specify number of packets to be transmitted or received in a MAC experiment instead of following time.

**My Address** - Specify the address of each node (between 1 and 6) in top layers. For example, the server uses this field in FTP to identify the client for sending and receiving files.

**Rx Mode** - Specify Promiscuous or Non-Promiscuous mode of packet reception. By default all the stand-alone experiments use promiscuous mode of reception (receives the entire packet). Altering this field will not have effect in the stand-alone experiments. Refer programmer's manual for using various types while writing codes for exercises.

**I/O Mode** - Fixes the types of transmit and receive modes. In majority of the stand-alone Experiments, blocking transmit and receive is used and the other types are left for the user to implement wherever necessary as suggested in exercises. Altering this field will not have effect in the stand-alone experiments. Refer programmer's manual for using various types while writing codes for exercises.

**Token Release Mode** - Specify the type of token release mode to be followed in token ring. Manual token release is used in token ring experiment. Altering this field will not have effect in this stand-alone experiment. Refer programmer's manual for using various types while writing codes for exercises.

**Direction** - Specify which direction to be followed by each node. In MAC experiments, Sender will only transmit data packets and Receiver will only receive. In DLL experiments, Sender will transmit data packets and receive acknowledgement packets while Receiver will receive them and transmit acknowledgement packets. In FTP Sender is the client and Receiver is the server.

**Boot File Name** - Name and path of the NIU driver file. This is for future use. The default file and location is C:\Lantrain\bin\LantV13.exe. It may be left to this default name.