East West University

Diminution of contention period of Carrier-sense multiple access with collision detection (CSMA/CD)

Submitted by:

Salma Akter

ID:2013-3-60-023

Tamera Tasnim

ID:2013-3-60-044

Hasina SharminJui

ID:2012-3-60-034

Supervised by:

Dr. Anisur Rahman

Assistant Professor

Department of Computer Science & Engineering (CSE)

East West University

A thesis submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering to the Department of Computer Science & Engineering (CSE)

December 2017

Abstract

Carrier- sense multiple access with collision detection (**CSMA/CD**) is a media access control method used most notably in early Ethernet technology for local area networking. It uses a carrier -sensing scheme in which a transmitting station detects collisions by sensing transmissions from other stations while transmitting a frame. When this collision condition is detected, the station stops transmitting that frame, transmits a jam signal, and then waits for a maximum time interval before trying to resend the frame. Our aim to reduce the propagation time less than 2T.In this thesis try to every node or transmitter can not wait 2T and every node or transmitter there have own sequence time to transmit.

Declaration

We, hereby, declare that this thesis under the supervision of Dr. Anisur Rahman, Assistant Lecturer, Department of Computer Science and engineering, East West University, Dhaka, Bangladesh. We also declare that no part of this thesis has been or is being submitted elsewhere for the award of any degree or diploma.

……...

Salma Akter

ID:2013-3-60-023

……

Tamera Tasnim

ID: 2013-3-60-044

……

Hasina SharminJui

ID:2012-3-60-34

Letter Of Acceptance

I hereby declare that, this thesis is the student's own work and best effort of mine. All other sources of information used have been acknowledged. This thesis has been submitted with our approval.

……

Dr. Anisur Rahman **Supervisor**

Assistant Professor

Department of Computer Science & Engineering (CSE)

East West University

……

Dr. Ahmed Wasif Reza **Chairperson**

Associate Professor

Department of Computer Science & Engineering (CSE)

East West University

Acknowledgement

First of all, we would like to thank almighty Allah for giving us strength, patience and knowledge to complete this project work.

We would like to thank our supervisor Dr. Anisur Rahman for his almost direction, constant help, support and feedbacks without this we would not have been able to successfully complete our thesis. He has been a source of inspiration for our throughout the process of the research and writing. His feedbacks and insights were always valuable, and never went unused.

Our deep gratefulness to all of our teachers, who have trained us at East West University. Their wonderful teaching methods improved our knowledge of the individual subject and enabled us to complete our studies in time.

We would like to thank our friends and seniors for their motivation and corporation.

This acknowledgment would not complete without thanking our parents for their endless support, encouragement during studies, great contribution and perfect guidance from the beginning to end.

Abbreviation and Acronyms

CA	Carrier-sense
MA	Multiple access
CSMA	Carrier-sense multiple access
CSMA/CD	Carrier-sense multiple access/collision detection
CSMA/CA	Carrier-sense multiple access/collision allocation
CSMA/CN	Carrier-sense multiple access/ collision notification
TCP/IP	Transmission Control Protocol/Internet Protocol
NIC	Network Interface Cards
MAC	Medium Access Control
IFG	Inter Frame Gap
OSI	Open System Interconnection
MAC	Medium Access Control

Table Of Contents

List of Tables

List of Figures

Chapter 1

Introduction

CSMA is a network access method used on shared network topologies such as Ethernet to control access to the network. Devices attached to the network cable listen CA (carrier sense) before transmitting. If the channel is in use, devices wait before transmitting. MA (multiple access) indicates that many devices can connect to and share the same network. All devices have equal access to use the network when it is clear. Even though devices attempt to sense whether the network is in use, there is a good chance that two stations will attempt to access it at the same time. On large networks, the transmission time between one end of the cable and another is enough that one station may access the cable even though another has already just accessed it. There is a method for avoiding these collision is called CSMA/CD [13].

The Carrier Sense Multiple Access/Collision Detection (CSMA/CD) protocol is a fundamental distributed protocol for networks. In CSMA/CD two or more stations share a common bus transmission medium. Indeed, it is one of the most important part of the IEEE 802.3 international standard (Ethernet Network Communication protocol).Ethernet is most widely installed local area network technology. Ethernet is a link layer protocol in the TCP/IP stack, describing how networking devices can format data for transmission to other network devices on the same network segment and how to put that out on the network connection. The Internet protocol suite is the conceptual model and set of communications protocols used on the internet and similar computer networks. It is commonly known as TCP/IP because the foundational protocols in the suite are the transmission Control Protocol(TCP) and the Internet Protocol(IP). In Ethernet, multiple Network Interface Cards (NIC) may be connected via the same channel. Since two NICs may send packets simultaneously, collisions may occur, thus discarding bothpackets. Both the NICs will detect this collision, but cannot re-send the packets at once, since it would induce a new collision. So, when a collision happens, the CSMA/CD protocol forces each NIC to pick at random an integer-valued delay from a bounded interval, and to wait for a length of time proportional to this integer-valued delay before re-sending the packet.[12]

1.1 Motivation

In Carrier Sense Multiple Access/Collision Detection (CSMA/CD) protocol is sending frame and wait the Acknowledgement (ACK). When a node wait for its ACK that time is called contention period . In CSMA/CD the contention period is 2τ.The node wait until 2τ.This paper concerned to reduce the propagation time. All the nodes there have individual table for transmission. In this table nodes have generate the sequence of transmission. When one node is transmit a frame it cannot wait until 2τ it waits less than 2τ time. We will discuss details in the paper about how and how much time can be.

1.2 Objective of the Research

CSMA/CD reduce the propagation time .The traditional CSMA (Carrier Sense Multiple Access/Collision Detection) the propagation time is 2T and every node or transmitter they have no individual propagation time. In our research, we work reduce the propagation time less than 2Tand every node have individual table for propagation time to transmit frame or data. Table have generate the sequence of transmission. In this research we try to compare our research with Carrier Sense Multiple Access/Collision Detection (CSMA/CD).

1.3 Outline of the report

The outline of the rest of this report has been structured as follow:

Chapter 2: Presents the background study of the project.

Chapter 3: Proposed Protocol.

Chapter 4: Experimental Results and Analysis.

Chapter 5: Outlines conclusion of this work precisely.

Chapter 2

Background Study

Networks have been divided into 7 layers where Data Link Layer is also one of them.MAC Layer is a sub-layer of the data link layer. CSMA protocols is one of the protocol of MAC Layer (Medium Access Layer) and are used by it. CSMA is based on the principal “Listen before talk”. It can reduce the probability of collision but cannot eliminate it.

After 1970s a revolutionary change was noticed in the world of computing. A major part of this revolutionary change in the use of computers has been the use of Ethernet LANs to enable communication among computers. Bob Metcalfe's 1973 Ethernet memo describes a networking system based on an earlier experiment in networking called the Aloha network. He developed a new system that included a mechanism that detected when a collision occurred (collision detect). The system also included "listen before talk," in which stations listened for activity (carrier sense) before transmitting and supported access to a shared channel by multiple stations (multiple access). Putting all these components together we can see why the Ethernet channel access protocol is called Carrier Sense Multiple Access with Collision Detect (CSMA/CD). [1]

2.1 Network Architecture

The main concept of the ISO-OSI layered architecture of Networks is that the process of communication between two endpoints in a telecommunication, network can be divided into seven distinct groups of related functions, or layers depending on the complexity of the functionality each of these layers provide. Each communicating user or program is at a computer that can provide those seven layers of function. So in a given message between users, there will be a flow of data down through the layers in the source computer, across the network and then up through the layers in the receiving computer. The seven Open Systems Interconnection layers are:

1. Physical Layer

2. Data Link Layer

3. Network Layer

4. Transport Layer

5. Session Layer

6. Presentation Layer

7. Application Layer. [2]

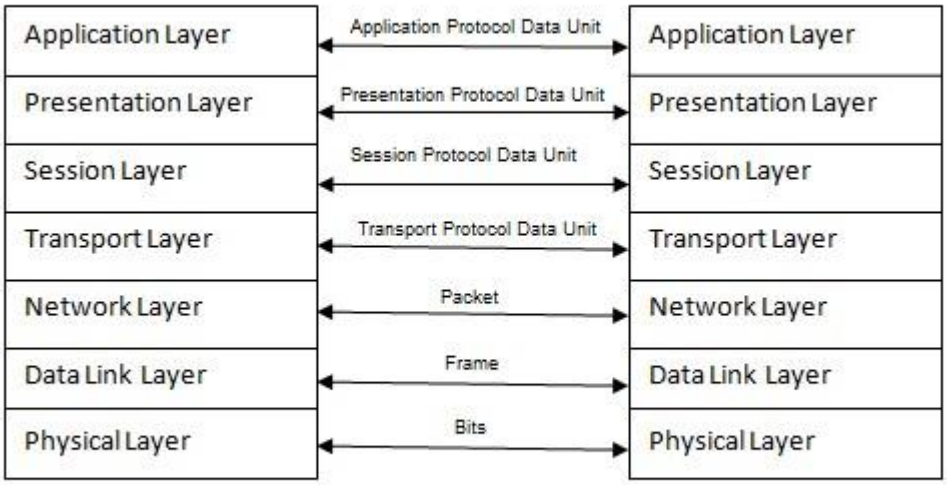

Figure 2.1 OSI Reference model

2.2 Data Link Layer

According to the ISO standards, networks have been divided into 7 layers depending on the complexity of the functionality each of these layers provide. Data link layer is the second layer in OSI reference model and lies above the physical layer. Data link layer is most reliable node to node delivery of data. It forms frames from the packets that are received from network layer and gives it to physical layer. It also synchronizes the information which is to be transmitted over the data. Error controlling is easily done.
The encoded data are then passed to physical.

Error detection bits are used by the data link layer. It also corrects the errors. Outgoing messages are assembled into frames. Then the system waits for the acknowledgements to be received after the transmission. It is reliable to send messages. The data link layer performs the following functions. This layer provides reliable transmission of a packet by using the services of the physical layer which transmits bits over the medium in an unreliable fashion. This layer is concerned with -

- **Framing:** Breaking input data into frames (typically a few hundred bytes) and caring about the frame boundaries and the size of each frame.
- **Acknowledgment:** Sent by the receiving end to inform the source that the frame was received without any error.
- **Sequence Numbering:** To acknowledge which frame was received.
- **Error Detection:** The frames may be damaged, lost or duplicated leading to errors. The error control is on link to link basis.
- **Retransmission:** The packet is retransmitted if the source fails to receive acknowledgment.
- **Flow Control:** Necessary for a fast transmitter to keep pace with a slow receiver.

The data link layer is divided into two sub layers: The Media Access Control (MAC) layer and the Logical Link Control (LLC) layer. The MAC sub layer controls how a computer on the network gains access to the data and permission to transmit it. Within the semantics of the OSI network architecture, the data-link-layer protocols respond to service requests from the network layer and they perform their function by issuing service requests to the physical layer. [3]

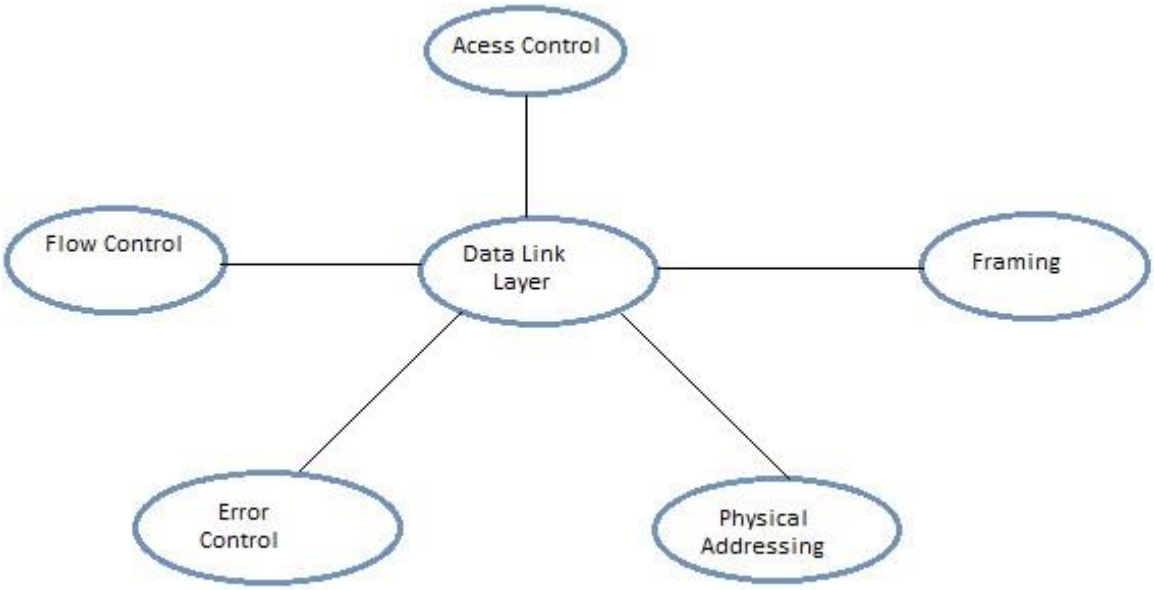

Figure 2.2 Functions of Data Link Layer

2.3 MAC (Media Access Control) Layer

The Media Access Control (MAC) data communication Networks protocol sub-layer, also known as the Medium Access Control, is a sub-layer of the data link layer specified in the seven-layer OSI model. The medium access layer was made necessary by systems that share a common communications medium. Typically these are local area networks. The Media Access Control sub layer also determines where one frame of data ends and the next one starts. There are four means of doing that: a time based, character counting, byte stuffing and bit stuffing. The MAC layer is the "low" part of the second OSI layer, the layer of the "data link". The primary functions performed by the MAC Layer are-

- Frame delimiting and recognition.
- Addressing of destination stations (both as individual stations and as groups of stations).
- Conveyance of source-station addressing information.
- Transparent data transfer of LLC PDUs, or of equivalent information in the Ethernet sub-layer.
- Protection against errors, generally by means of generating and checking frame check sequences and control of access to the physical transmission medium. [4]

MAC may refer to the sub layer that determines who is allowed to access the media at any one time (e.g. CSMA/CD). Other times it refers to a frame structure delivered based on MAC addresses inside.

There are generally two forms of media access control: distributed and centralized. Both of these may be compared to communication between people. In a network made up of people speaking, i.e. a conversation, we look for clues from our fellow talkers to see if any of them appear to be about to speak. If two people speak at the same time, they will each pause a random amount of time and then attempt to speak again, effectively establishing a long and elaborate game of saying "no, you first".

The Media Access Control sub layer also determines where one frame of data ends and the next one starts – frame synchronization. There are four means of frame synchronization: time based, character counting, byte stuffing and bit stuffing.

1. Time Based:

The time based approach simply puts a specified amount of time between frames. The major drawback of this is that new gaps can be introduced or old gaps can be lost due to external influences.

2. Character Counting:

Character counting simply notes the count of remaining characters in the frame's header. This method, however, is easily disturbed if this field gets faulty in some way, thus making it hard to keep up synchronization.

3. Byte Stuffing:

Byte stuffing precedes the frame with a special byte sequence such as DLE STX and succeeds it with DLE ETX. Appearances of DLE (byte value 0x10) have to be escaped with another DLE. The start and stop marks are detected at the receiver and removed as well as the inserted DLE characters.

4. Bit Stuffing:

Similarly, bit stuffing replaces these start and end marks with flag consisting of a special bit pattern (e.g. a 0, six 1 bits and a 0). Occurrences of this bit pattern in the data to be transmitted are avoided by inserting a bit. To use the example where the flag is 01111110, a 0 is inserted after 5 consecutive 1's in the data stream. The flags and the inserted 0's are removed at the receiving end. This makes for arbitrary long frames and easy synchronization for the recipient. Note that this stuffed bit is added even if the following

data bit is 0, which could not be mistaken for a sync sequence, so that the receiver can unambiguously distinguish stuffed bits from normal bits.

2.4 ALOHA

In 1970s Norman Abramson proposed a new and reliable algorithm to solve the channel allocation problem in wired network. Abramson worked with his colleagues at the University of Hawaii to develop this method called ALOHA or Pure ALOHA. It's another version is called Slotted ALOHA.

Aloha is a simple communication protocol where each source in the network transmits data whenever it has a frame to be transmitted. If the frame is transmitted successfully, the next frame will be transmitted. If the transmission is failed, the source will send the same frame again. Aloha works well with wireless broadcast systems or half-duplex two-way links. But when the network becomes more complex, such as an Ethernet with multiple sources and destinations that uses a common data path, problems occur due to colliding of data frames. When the communication volume increases, the collision problem becomes worse. This can reduce the efficiency of a network since colliding frames will cause loss of data in both the frames. Slotted Aloha is an improvement to the original Aloha protocol, where discrete time slots were introduced to increase the maximum throughput while reducing collisions. This is achieved by allowing sources to transmit only at the beginning of a timeslot.

2.4.1 Pure Aloha

Pure Aloha is a random access protocol. A user can access the channel whenever it has data to be transmitted. Definitely, there will be a collision. However, after transmission the user waits for an acknowledgment from separate feedback channel. If there is collision, the sender waits for a random amount of time and retransmits the data. Pure ALOHA does not relate to time synchronization.

2.4.2 Slotted ALOHA

Slotted Aloha divides the time into equal time slots of length greater than the packet duration. Each user has synchronized clock and transmits the data only at the beginning of new time slot. This helps in a discrete distribution of accessing the channel. But collision is not prevented absolutely; there is a collision with portions of data packets.[5]

2.5 CSMA (Carrier-sense multiple access)

Carrier sense multiple access (CSMA) algorithm is based on the concept that each station on the network is able to sense the channel before transmitting the data packet. Sensing the channel means to monitor the status of channel whether it is idle or busy. If the channel is idle/free, then station can transmit the data. But if the channel is sensed busy, the station will wait and keep on sensing the carrier till it becomes free. This method decreases the probability of collision.

There are several versions of CSMA exist:

1. **non-persistent:** In this type of CSMA, a station senses the channel first. If the channel is free then it starts transmission immediately. But if channel is busy then the station does not continuously sense the channel, rather it waitsfor a random amount of time and then repeats the algorithm.
2. **p-persistent:** It is applied to slotted channel. Here stations also sense the medium. If the medium is free, a station transmits the packet with aprobability of *p,* or with probability of *1-p* if the station defector next slot.

3. 1-persistent: When a station wants to send the data, it first senses to thechannel whether it is free or busy at the moment. If it is busy, the station waitsuntil it becomes free. And if the station detects an idle channel, it transmits adata frame. When the channel becomes free the two or more neighboring stations can transmit data at the same time. This will cause collisions. If thecollision occurs, the station waits a random amount of time and repeats themethod. The algorithm is called 1-persistent because the station transmits witha probably of 1 whenever it finds an idle channel.[6]

2.6 CSMA/CD (Carrier-sense multiple access/collision detection)

CSMA/CD is a modification of pure carrier-sense multiple access (CSMA). CSMA/CD is used to improve CSMA performance by terminating transmission as soon as a collision is detected, thus shortening the time required before a retry can be attempted. Carrier Sense Multiple Access/Collision Detect (CSMA/CD) is the protocol for carrier transmission access in Ethernet networks. Ethernet local area network (LAN) uses CSMA/CD. The Ethernet network may be used to provide shared access by a group of attached nodes to the physical medium which connects the nodes. These nodes are said to form a Collision Domain. All frames sent on the medium are physically received by all receivers, however the Medium Access Control (MAC) header contains a MAC destination address which ensure only the specified destination actually forwards the received frame (the other computers all discard the frames which are not addressed to them). CSMA/CD is a protocol in which the station senses the carrier or channel before transmitting frame just as in persistent and non-persistent CSMA. On Ethernet, any device

can try to send a frame at any time. Each device senses whether the line is idle and available to be used. If it is, the device begins to transmit its first frame. If another device has tried to send at the same time, a collision is said to occur and the frames are discarded. Each device then waits a random amount of time and retries until successful in getting its transmission sent. In a collision, the issuer immediately cancel the sending of the package. This allows to limit the duration of collisions. After a collision, the transmitter waits again silence and again, he continued his hold for a random number; but this time the random number is nearly double the previous one: it is this called back-off exponential. In fact, the window collision is simply doubled (unless it has already reached a maximum). From a packet is transmitted successfully, the window will return to its original size. CSMA /CD operates by detecting the occurrence of a collision. CSMA /CD takes effect after a collision and minimizes the recovery time.[7]

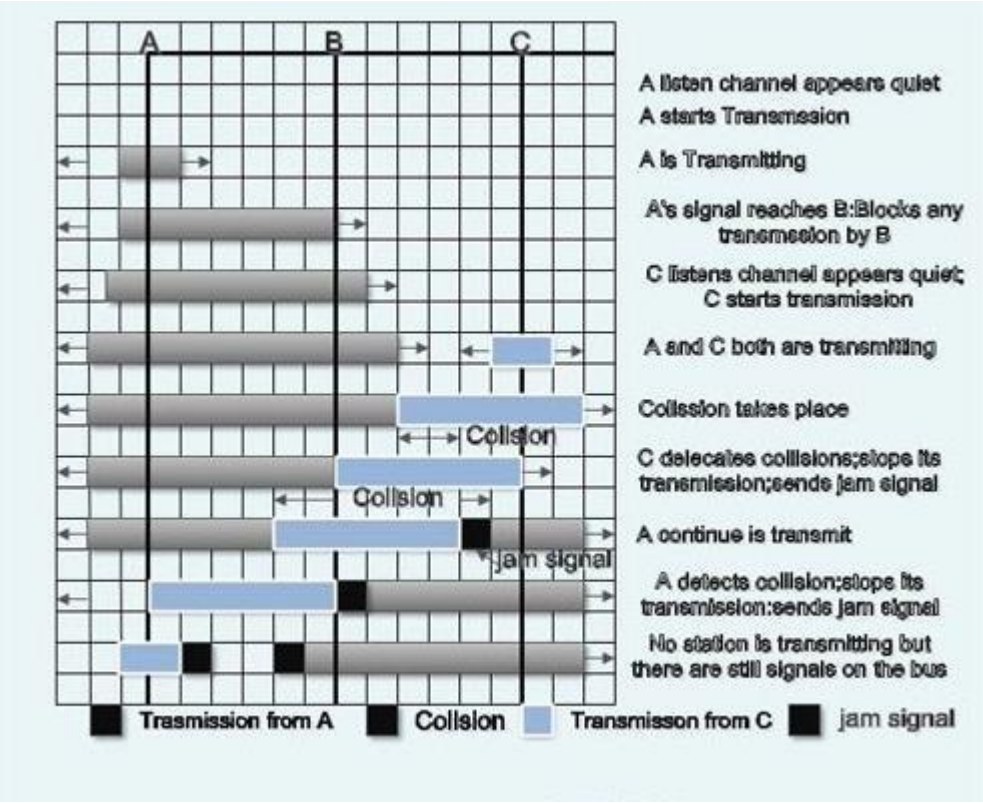

Figure 2.3: Transmission of CSMA/CD

2.6.1 Collision Mechanism

Devices attached to the network cable listen (carrier sense) before transmitting. All devices have equal access to use the network. The station senses the carrier or channel first. Each device senses whether the line is idle .If no transmission is taking place at the time, the particular device can transmit. If two stations

attempt to transmit simultaneously, this causes a collision. In a collision, the issuer immediately cancel the sending of the package. This allows to limit the duration of collisions. As thus, time doesn't waste to send a packet complete if it detects a collision. After a collision, the transmitter waits again and again. It continued his hold for a random number. But this time the random number is nearly double the previous one: it is this called back-off exponential algorithm. In fact, the window collision is simply doubled (unless it has already reached a maximum). From a packet is transmitted successfully, the window will return to its original size.

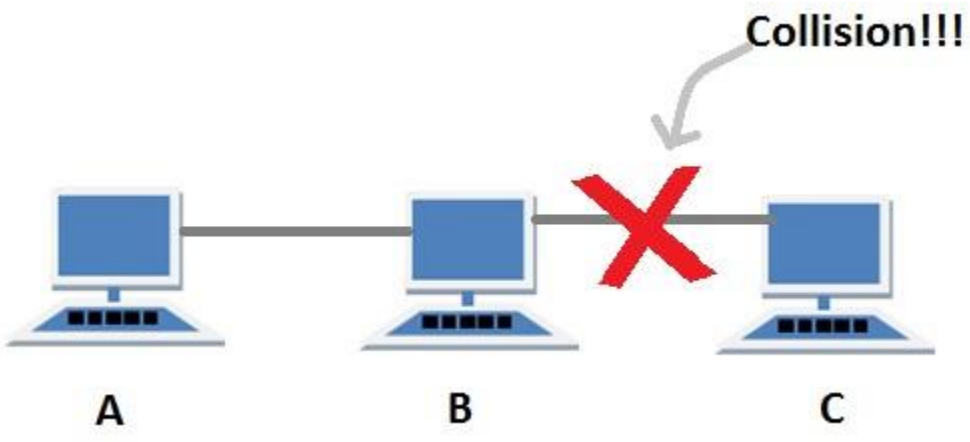

Figure 2.4 Collision between Two Nodes

If the channel is busy, the station waits. And it listens at the same time on communication media to ensure that there is no collision with a packet sent by another station.

2.6.2 Contention Period

Contention means competition for resources. The term is used especially in networks to describe the situation where two or more nodes attempt to transmit a message across the same wire at the same time. It is a type of network protocol that allows nodes to contend for network access. That is, two or more nodes may try to send messages across the network simultaneously. The contention protocol defines what happens when this occurs. The most widely used contention protocol is CSMA/CD, used by Ethernet. [8]

2.6.3 Random Back-off time / Binary Exponential Back-off

Binary exponential back-off refers to an algorithm used to space out repeated retransmissions of the same block of data, often as part of network congestion avoidance.

Examples are the retransmission of frames in carrier sense multiple access with collision detection (CSMA/CD) networks, where this algorithm is part of the channel access method used to send data on these networks. In Ethernet networks, the algorithm is commonly used to schedule retransmissions after collisions. The retransmission is delayed by an amount of time derived from the slot time and the number of attempts to retransmit.

This is the mostly used algorithm in order to select the random amount for the duration of waiting time in the network. Here the random amount of time is the random back-off time that counts downwards to zero. This time delays the access of medium in order to provide transparent and collision free environment for all nodes in the network. Whenever, if any node finds the medium busy in the network, it is supposed to get a random value within a contention window for back-off time. The node starts counting down its back-off time only when the medium becomes free. Each node may have different or same amount of time but within contention window. This random waiting time avoids collisions; otherwise all nodes would have accessed the idle medium at the same time. After finishing that random time, they start sensing the medium. As soon as a node senses the channel is busy, it loses this turn and it will select another back-off time for the next cycle. On the other hand, if a node gets the medium free after waiting for random time, it can access the medium immediately.[9]

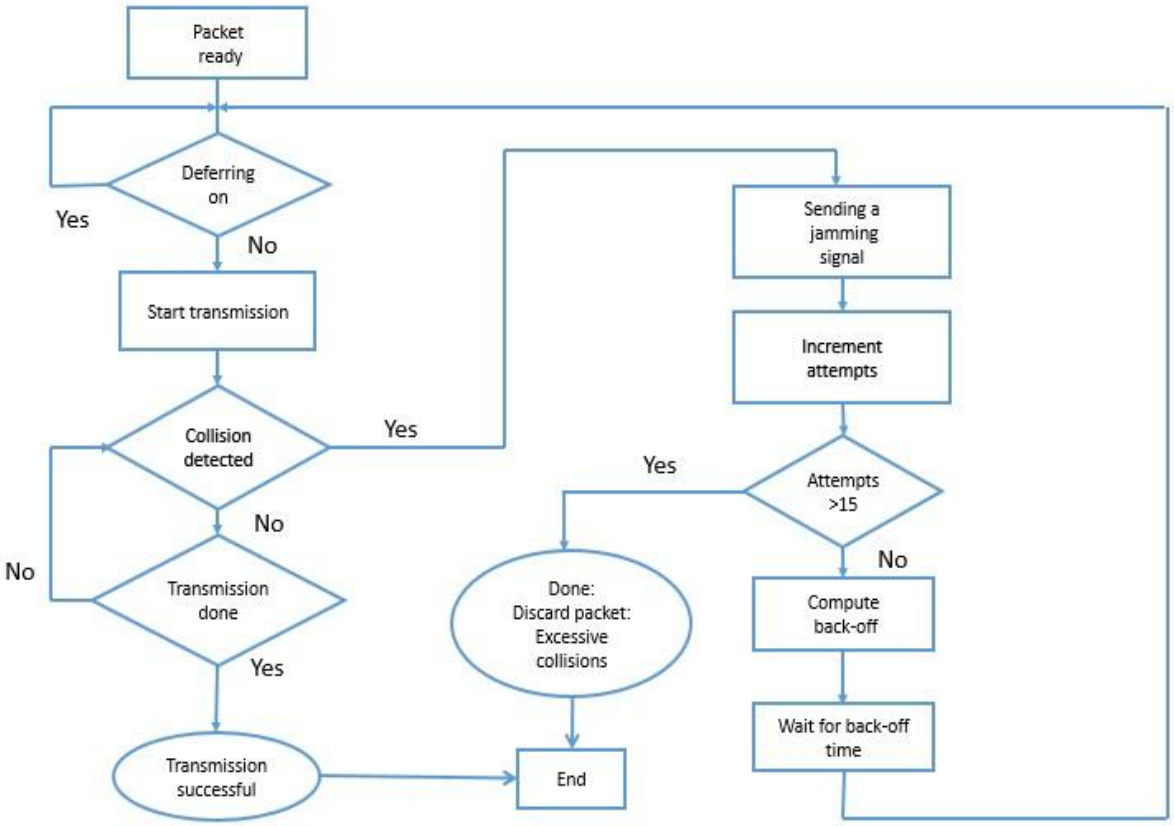

Figure 2.5 Binary Exponential Back off flowchart

2.7 CSMA/CA (Carrier-sense multiple access/collision avoidance)

CSMA /CA operates by sensing the state of the medium in order to prevent or recover from a collision. A collision happens when two transmitters transmit at the same time. Wireless LAN uses CSMA/CA. CSMA CD only minimizes the recovery time. This is the CSMA protocol with collision avoidance. When the station ready to transmit, senses the line by using one of the persistent strategies. As soon as it find the line to be idle, the station waits for an IFG (Inter frame gap) amount of time. Then waits for some random time and sends the frame. After sending the frame, it sets a timer and waits for the acknowledgement from the receiver. If the acknowledgement is received before expiry of the timer, then the transmission is successful. But if the transmitting station does not receive the expected acknowledgement before the timer expiry then it increments the back off parameter, waits for the back off time and re-senses the line.

CSMA /CA operates by sensing the state of the medium in order to prevent or recover from a collision. CSMA /CA takes effect before a collision .CSMA/CA reduces the possibility of a collision and somewhat increase bandwidth utilization.[10]

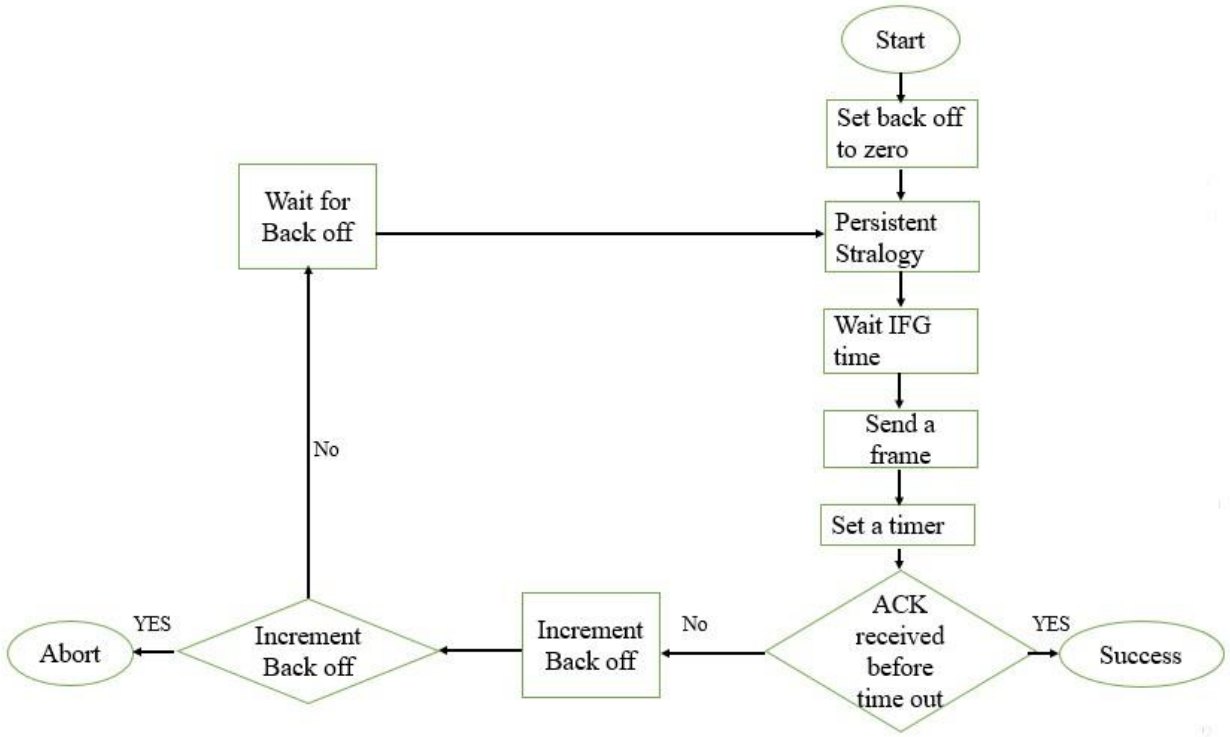

Figure 2.6 CSMA/CA Flowchart

2.8 CSMA/CN (Carrier-sense multiple access/collision notification)

CSMA/CN is an early attempt to rethink medium access control in wireless networks. The operation of CSMA/CN can be summarized as follows. Under CSMA/CN, the receiver uses PHY layer information to detect a collision and immediately notifies the transmitter. The collision notification consists of a unique signature, sent on the same channel as the data. The transmitter employs a listener antenna and performs signature correlation to discern this notification. Once discerned, the transmitter immediately aborts transmission.

The transmitter has two interfaces tuned to the same channel, one for transmission and another for listening. The receiver has a single interface (Figure 1). Once communication begins, the receiver exploits preamble correlation to detect the presence of an interference. Realizing that the packet is likely to fail, the receiver checks the confidence of incoming bits via physical layer hints from SoftPHY [2, 1]. When

the receiver is reasonably confident of an error, it initiates a collision notification to the transmitter. The notification is a short signature unique to the receiver, also known to the transmitter. The transmitter's listening antenna continuously "searches" for this signature using correlation. Even in the presence of a strong signal from the transmit antenna, signature correlation at the listening antenna can reliably discern the collision notification. The transmitter aborts, releasing the channel for nearby transmitters.

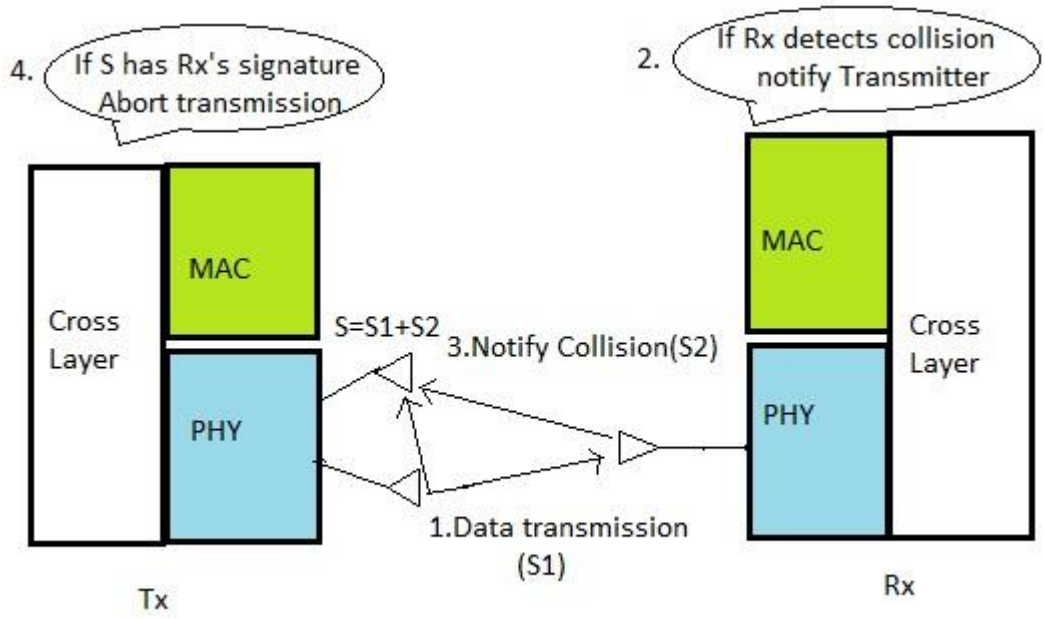

Figure 2.7 Basic Structure of CSMA /CN

Wireless broadcast/multicast protocols traditionally suffer from the problem of excessive ACK overhead. CSMA/CN may resolve this problem if unique signatures can be assigned to each of the clients.
CSMA/CN is only one example of how signal correlation can be exploited in wireless systems.[11]

Chapter 3

Proposed Protocol

This chapter introduces the concept of a CSMA/CD and flow chart of CSMA/CD. Then we discuss about proposed part of CSMA/CD Proposed part of CSMA/CD are elaborate here. This chapter also compare between CSMA/CD and proposed part of CSMA/CD.

3.1 CSMA/CD

The acronym CSMA/CD signifies carrier-sense multiple access with collision detection and describes how the Ethernet protocol regulates communication among nodes. CSMA/CD is a type of contention protocol. Networks using the CSMA/CD procedure which is simple to implement. It works by sensing the carrier first. The mechanism of collision detection which CSMA/CD follows is through listening while talking. What this means is so long as a node is transmitting the packet, it is listening on the cable. Carrier Sense Multiple Access / Collision Detection determines how network devices respond when two devices attempt to use a data channel simultaneously. Actually CSMA/CD is the most widely used protocol for determining how network devices respond in the event of a collision. Standard Ethernet networks use CSMA/CD to physically monitor the traffic on the line at participating stations. If no transmission is taking place at the time, the particular station can transmit and wait for successful transmission. If two stations attempt to transmit simultaneously, this causes a collision. When the collision is detected, they stop their transmission immediately .Then Send a jamming signal on the line to warn all nodes that a collision has happened and they should not transmit .That means they need to back off from transmission at this moment. However two nodes whose signal has created collision cease transmitting and wait a random amount of time and retries until successful in getting its transmission sent. 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. As CSMA/CD is simple to implement, this has helped make it an international standard and an important part of the Ethernet, which is the most widely deployed architecture for LANs.

3.1.1 Binary Exponential Back Off

In back off procedure, at the beginning of each slot, a station transmits if its back off timer has expired by that time. Otherwise, depending on the channel state (idle or busy), by end of the current slot, the station will count down the back off counter by 1 or will have frozen it at a value. Suppose the current slot ends in the busy state. When next slot begins, the stations naturally fall into one of the two groups: those that 4 have transmitted, and those who did not and have frozen their back-off counters. Ideally the back off counter does not freeze at 0. This introduces a short-term unfairness among the two groups of stations: only stations in the former group have privileged possibility to access this slot since after transmitting these stations generate a new back off time with a chance to be 0. For convenience, we refer a slot subsequent to a busy slot as a post-busy slot. This subtle yet crucial different access behavior in the post busy slot leads the back off algorithm deviate the p-persistent model commonly assumed in previous work**s.**

3.2 CSMA/CD Flow Chart

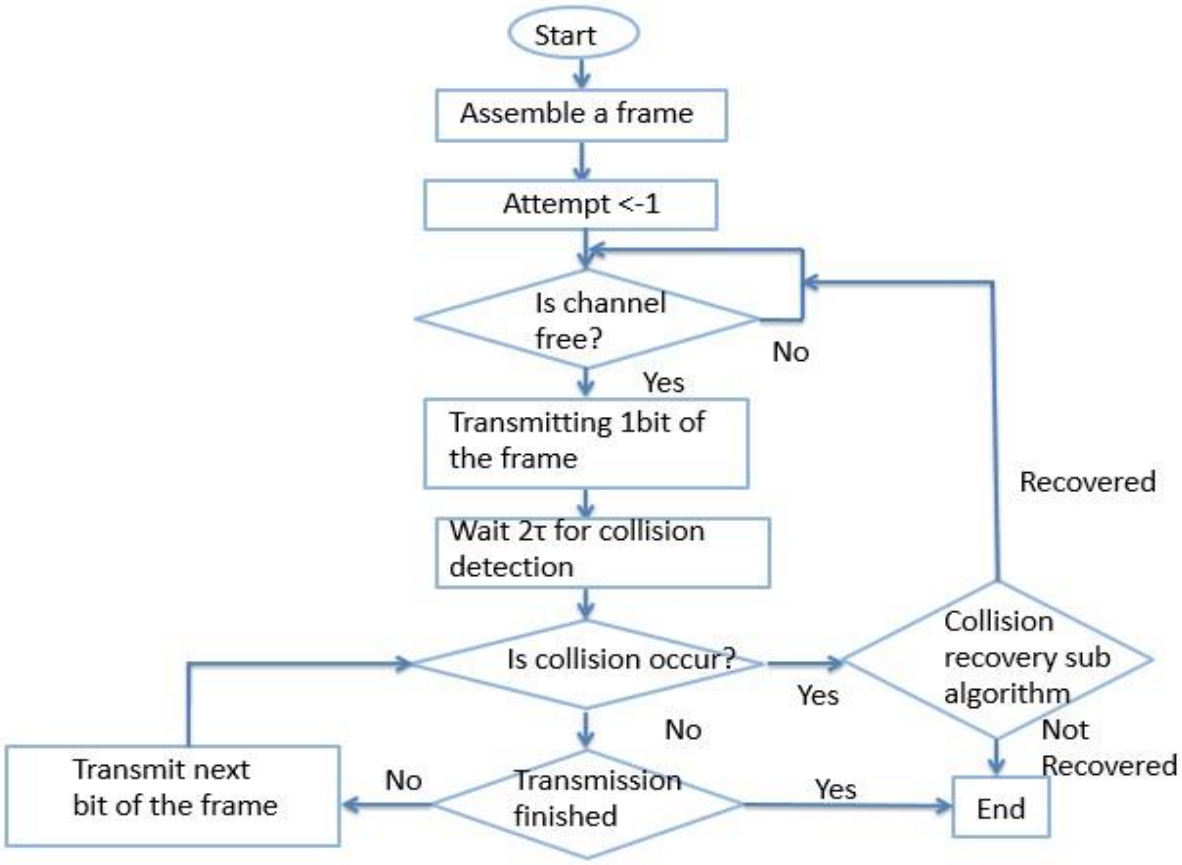

Figure 3.1 CSMA/CD flowchart

3.3 Pseudo code of CSMA/CD

----------Pseudo code----------

CSMA/CD

Start

Z[100]={ } **/1D array for keep time for each transmission**

int r=0 int p=0 int c=0 int data1=0 int data2=0 do until

data1=50

do until data2=50 q=randomly generate integer value less than or equal 100

n[data1,data2]=q **/randomly generate integer value for distance between two nodes**

loop end

loop end inti=0 int

m=0 int k=0 int

status[50,50]={ }

int a int b int c do

until i=100

do until m=50

do until k=50 m1=randomly generate integer value less than 3

status[m,k]=m1 **/randomly update the status of each channel, channel free or busy**

loop end loop end a= randomly generate integer value less than or

equal 100 **/select sender randomly** b= randomly generate integer value less than or

equal 100 **/select sender randomly**c=2*100; **/Maximum distance between two**

node is 100 nano seconds

if status(a,b)==1 and status(a,b)==1 **/when both node are free**

start time count

pause(c) end time

count z(i)=counted time

```
/There is no transmission
        End if
        elseif status(a,b)==2 and status(a,b)==2  /when both node are busyz(i)=0;
       /There is no transmission
        End elseif else
       /there occurs a collision          start
time count

                pause(c)

end time count          z(i)=counted time

/Number of collision              p=p+1;
         End if
        r=r+z(i); /Total  time
End if
Display r
Display p
End
```

3.4 Proposed Enhanced CSMA/CD

CSMA/CD is a protocol in which the station senses the carrier or channel before transmitting frame. In CSMA/CD contention time is 2τwhere τ is the maximum distance time or propagation time among the all nodes. So if any transmission is going on the medium then other stations have to wait for 2τ time. If collision happens, then it will happen within double of the propagation time between two nodes. Even if the propagation time between two nodes is less than τ, even then the sender and receiver have to wait 2τ time in CSMA/CD. The nodes wait for 2τ time to detect if the collision occurs or not. But then time wastes. So we reduce the waiting period for detecting the collision. Here, in our project, we proposed a method of reducing this contention time.

Now, we give a detailed description about the process of decreasing contention period in proposed CSMA/CD.

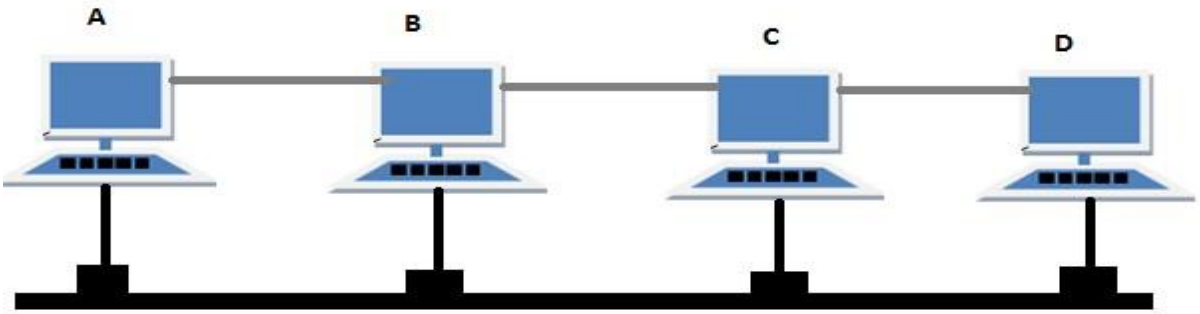

Figure 3.2 Transmission among four nodes

Our proposed method shows how to reduce this contention time. As our method, if node A want to transfer data to node B, then our proposed contention time is $2\tau_{AB.}$ They do not need to wait for 2τ time.

For node A and B time, contention period is $2\tau_{AB}$, which is less than or equal to τ. In the $2\tau_{AB}$ time, the nodes detect the collision happens or not. For this, we have to calculate the propagation time for each node. However, each node have to well known about the propagation time distance for every node. Also So now nodes do not need to wait for 2τ time. Nodes can wait for according to their own contention time which is double to the propagation time between them. Because if there occurs a collision in the midway in the channel , the damaged frame will come back to sender and send can know about the collision.

That's why, the contention period is used or the nodes have to wait for the collision detection and they stop transition until the contention period is over. So Suppose there are five nodes A,B,C and D. So, if node A want to transmit a frame to node D.

	A	B	C	D
A	0	4	8	21
B	4	0	30	4
C	8	30	0	7
D	21	4	7	0

Table 3.1 Example of transmit data for four nodes (A to D)

So, using our method, A will wait for 21 nanoseconds, before it have to wait for 30nanoseconds, which is maximum propagation time in the LAN.

3.4 Proposed Enhanced CSMA/CD Flow Chart

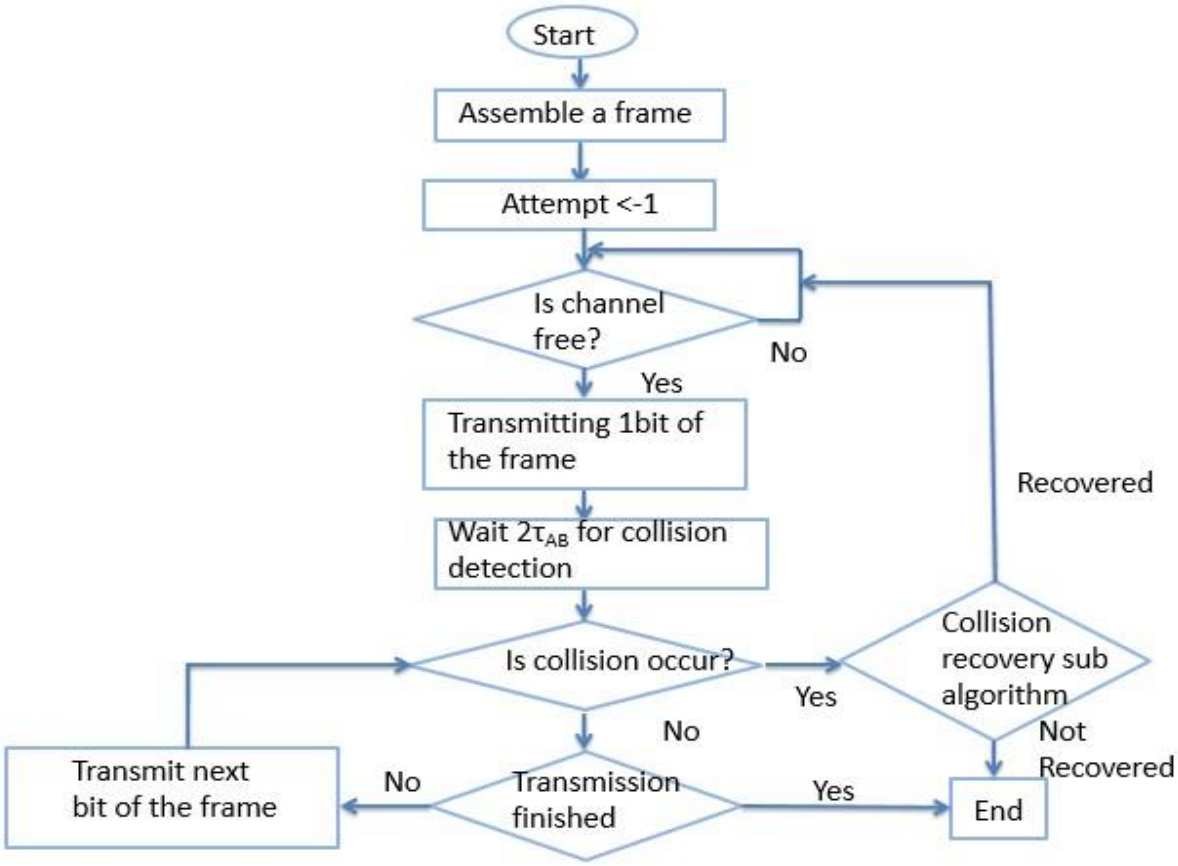

Figure 3.3: Proposed Enhanced CSMA/CD flowchart when node A transmit to node B

3.6 Pseudo code of Proposed Enhanced CSMA/CD

----------Pseudo code----------

Proposed Enhanced CSMA/CD

Start

Z[100]={ } **/1D array for keep time for each**

transmissionint r=0 int p=0 int c=0

```
int data1=0 int
data2=0 do until
data1=50
do until data2=50                    q=randomly generate integer value less than or equal 100
n[data1,data2]=q  /randomly generate integer value for distance between two nodes
          loop end
loop   end   inti=0   int
m=0   int      k=0   int
status[50,50]={       }
int  a  int  b  int  c  do
until i=100
do until m=50    do until k=50                m1=randomly generate integer value less than 3
status[m,k]=m1  /randomly update the status of each channel, channel free or busy
                    loop end loop end a= randomly generate integer value less than or
equal 100        /select sender randomly b= randomly generate integer value less than or
equal 100      /select sender randomly c=2*n(a,b); /n(a.b) is the distance time
between two nodes in nano second if status(a,b)==1 and status(a,b)==1 /when both
node are free
          start time count
pause(c)           end time
count    z(i)=counted time
/There is no transmission
          End if
          elseif status(a,b)==2 and status(a,b)==2 /when both node are busy
z(i)=0;
         /There is no transmission
          End elseif else
         /there occurs a collision
                    start time count
```

```
                pause(c)

end time count          z(i)=counted time

/Number of collision          p=p+1;
          End if
         r=r+z(i); /Total  time
End if
Display r
Display p
End
```

Chapter 4

Experimental Results and Analysis

This chapter simulates the CSMA/CD and proposed method for reducing the contention period of CSMA/CD. The experimental designed based on 50 nodes and 100 transactions among the nodes. Every node has distance time with each other in nanoseconds. Here distance time means the time of propagation time between two nodes. The size of the LAN is 30 meter. All the nodes of the LAN are into 30 meter. So, the maximum distance time is 100 nanoseconds where we know, radio signal can transmit 1 meter in 0.3nanoseconds.

To simulate the proposed method, we have calculated the total contention period time and number of collision for 100 transmissions among the nodes. In the LAN, the nodes are able to transmit for randomly if the channel is free and then wait for a fixed time which is called contention period.

In CSMA/CD, the nodes have to wait 2τ time where τ is the maximum distance time or propagation time among the all nodes. If the propagation time between two nodes is less than τ but the sender and receiver have to wait 2τ time for detecting the collision is occurred or not. This is the wastage of time, so we reduce the waiting period for detecting the collision.

In our method, the nodes have not to wait for 2τ time. If node A transmits a frame to node B, then they need to wait for $_B$ which is less than or equal to τ. In the $_B$ time, the nodes detect the collision is occurs or not. For this, we have to calculate the propagation time for each node.

For each transmission, the channel will be updated the automatically. Then it will check the channel is free or not. If the both node (sender and receiver) are free then there will be occurred a collision and wait contention period. If the both node (sender and receiver) are busy, there is no transmission occurred. Otherwise, there is a collision and the nodes wait for contention period.

When 50 nodes want to transmit 100 data frames using CSMA/CD protocol. The nodes have to wait contention period after send a data frame. Here, given contention period for every transmission. Even if, there is occurred a collision, the nodes have to wait for contention period. Only for no transmission when the channel was busy, there is no contention period.

Here, The Propagation distance between two nodes for each nodes.

Here given , the propagation distance among 50 nodes to each other.

Table 4.1:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	82	91	13	92	64	10	28	55	96	97	16	98	96	49	81
2	28	68	66	17	12	50	96	35	59	23	76	26	51	70	90
3	17	80	32	53	17	61	27	66	69	75	46	9	23	92	16
4	42	5	91	95	50	49	34	91	37	12	79	39	25	41	10
5	65	38	82	54	36	94	88	56	63	59	21	31	48	24	85
6	97	55	53	24	49	63	68	40	37	99	4	89	92	80	10
7	6	69	5	8	53	10	82	82	73	15	66	52	98	65	81
8	31	71	67	54	70	67	18	13	100	18	4	57	89	67	20
9	43	10	60	48	70	70	64	4	7	32	54	66	41	82	72
10	92	1	47	43	47	78	33	79	48	4	18	73	48	16	35
11	59	55	87	27	32	12	94	65	48	64	55	65	55	73	53
12	74	40	69	71	45	2	34	43	28	20	83	43	89	40	77
13	86	57	93	70	59	82	88	99	1	87	62	99	53	48	81
14	15	19	5	64	29	54	70	50	54	45	13	50	86	88	28
15	56	86	35	45	6	18	67	34	90	12	99	54	71	100	29

Table 4.1 Propagation Distance of transmit data between (1-15) to (1-15)

Table 4.2:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	46	11	100	34	30	7	30	5	51	77	64	9	9	78	91
17	69	14	73	12	12	65	33	66	75	59	75	24	74	98	87
18	11	94	19	27	80	49	77	40	28	4	68	43	46	61	6
19	99	18	26	40	8	69	41	99	41	63	16	39	17	76	88
20	17	67	90	52	71	16	96	55	68	4	81	75	13	53	33
21	64	36	100	23	66	61	39	15	3	43	19	73	38	85	74
22	99	7	94	2	69	79	54	89	90	63	14	22	19	5	11
23	79	70	1	85	93	78	5	38	71	73	23	27	68	48	63
24	72	62	35	94	13	74	65	84	40	75	84	33	56	98	55
25	51	49	88	36	45	97	5	98	19	67	59	68	37	63	82
26	29	55	99	72	84	44	48	57	27	75	51	65	31	14	48
27	41	67	94	82	49	76	42	98	99	87	39	46	25	79	89
28	12	82	33	25	35	38	55	57	40	40	52	66	96	73	41
29	51	62	82	54	21	46	43	97	63	70	73	35	52	56	16
30	2	44	84	62	53	87	10	91	11	52	15	56	1	77	85

Table 4.2 Propagation Distance of transmit data between (16-30) to (1-15)

Table 4.3:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
31	15	27	9	43	26	30	43	12	50	71	25	79	8	40	1
32	32	61	92	91	60	34	86	45	91	4	54	72	18	34	19
33	68	44	70	26	1	54	28	95	91	40	3	68	84	98	6
34	65	23	84	98	85	51	28	75	24	96	63	61	18	10	26
35	84	59	95	7	59	29	83	20	45	40	83	68	21	32	14
36	81	36	8	60	92	20	44	75	4	95	77	56	19	50	52
37	20	20	33	89	48	41	18	97	41	85	62	38	88	79	47
38	25	85	86	97	49	23	23	54	77	35	47	64	92	17	72
39	20	76	35	42	16	82	63	74	81	7	96	50	76	75	84
40	53	33	84	82	56	27	69	24	46	39	54	100	76	99	24
41	75	90	25	13	23	36	29	93	6	60	17	84	17	51	100
42	40	60	81	11	83	85	36	44	58	71	75	76	39	43	96
43	99	56	94	73	49	64	89	20	40	100	41	66	91	100	66
44	33	64	24	58	61	60	45	4	52	41	11	46	46	56	81
45	6	82	53	70	22	55	71	96	45	9	6	63	80	70	35
46	15	59	8	83	73	93	50	66	90	54	29	98	4	33	98
47	11	86	70	74	66	52	33	67	12	15	2	97	98	13	47
48	92	51	98	20	12	30	40	43	32	70	10	41	30	31	11
49	82	19	13	83	64	2	90	52	55	61	77	86	39	9	74
50	91	42	36	49	26	93	47	26	44	71	41	19	86	59	38

Table 4.3 Propagation Distance of transmit data between (31-50) to (1-15)

Table 4.4:

	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	15	43	92	80	96	66	4	85	94	68	76	75	40	66	18
2	96	55	14	15	26	85	26	82	25	93	35	20	26	62	48
3	83	54	100	8	45	11	97	1	78	82	87	9	40	26	81
4	14	95	96	58	6	24	36	83	2	5	17	65	74	65	46
5	20	23	18	23	44	32	93	44	19	91	98	44	12	26	41
6	27	34	68	14	73	11	66	50	78	72	91	90	34	70	20
7	46	44	83	9	14	18	40	84	81	7	40	53	42	66	63
8	37	47	99	16	86	65	38	20	43	49	13	59	23	39	59
9	97	54	33	11	62	78	43	10	27	16	29	45	53	46	88
10	61	20	74	25	92	27	77	19	29	10	58	69	55	43	65
11	100	22	11	11	7	41	45	37	77	63	78	94	98	20	14
12	40	81	76	38	22	80	95	33	68	44	84	77	17	87	99
13	23	50	91	58	85	74	59	25	67	9	63	67	73	90	99
14	21	57	65	42	21	95	9	11	15	17	63	58	6	94	73
15	42	47	77	82	11	18	36	6	53	34	18	21	91	68	47
16	54	11	83	34	30	75	2	5	67	61	53	73	71	79	29
17	9	37	37	69	60	79	37	21	9	78	21	39	56	23	65
18	32	78	70	13	14	10	1	43	66	73	54	11	64	13	14
19	36	69	30	54	84	60	34	30	46	43	36	56	75	43	43
20	55	40	42	19	26	3	93	66	94	17	93	80	58	45	26

Table 4.4 Propagation Distance of transmit data between (1-20) to (16-30)

Table 4.5:

31	32	33	34	35	36	37	38	39	40
71	4	28	5	10	83	70	32	96	4
36	84	59	55	92	29	76	76	39	57
44	92	19	27	15	14	87	58	55	15
55	30	75	19	69	19	37	63	79	9
60	27	61	72	23	12	30	32	43	51
4	75	51	48	91	61	62	86	81	58
30	44	2	99	17	11	38	20	49	34
26	30	62	27	83	99	74	35	59	11
52	95	64	96	25	68	29	68	70	7
65	68	64	95	21	71	24	12	61	46
70	10	53	54	87	49	40	68	75	53
52	89	59	16	20	41	75	83	79	32
77	59	93	59	2	13	87	49	85	21
74	7	87	94	99	86	79	52	18	40
92	11	75	74	57	19	60	30	14	22
70	56	40	7	79	34	61	75	11	13
49	16	79	11	30	24	54	10	41	11
10	15	17	20	32	32	22	26	90	71
13	3	30	32	66	96	94	46	25	77
76	23	7	77	68	72	65	42	40	82

Table 4.5 Propagation Distance of transmit data between (1-20) to (31-40) Table 4.6:

	41	42	43	44	45	46	47	48	49	50
1	44	39	77	80	19	49	45	65	71	76
2	8	6	54	78	94	13	57	47	2	34
3	86	63	36	52	41	8	24	13	19	24
4	93	78	49	44	45	31	51	52	82	80
5	9	27	81	3	93	74	49	58	24	46
6	19	24	89	3	49	17	98	72	51	48
7	96	93	6	74	27	43	55	95	42	99
8	91	88	82	27	60	3	43	32	17	18
9	26	23	67	85	35	79	68	1	61	39
10	46	67	78	36	67	42	85	84	26	62
11	35	15	59	27	5	76	25	45	69	36
12	54	9	12	14	68	50	19	50	15	6
13	56	63	4	62	37	5	49	20	13	21
14	14	4	94	31	30	34	47	65	3	85
15	90	8	25	6	45	2	90	20	10	31
16	55	49	90	80	74	6	8	9	80	95
17	12	79	30	61	97	44	70	76	44	66
18	56	19	22	8	92	71	56	32	17	63
19	76	75	75	11	69	47	22	10	83	18
20	32	82	79	86	51	64	96	45	7	87

Table 4.6 Propagation Distance of transmit data between (1-20) to (41-50)

Table 4.7:

	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
20	55	40	42	19	26	3	93	66	94	17	93	80	58	45	26
21	58	18	96	27	93	23	38	9	65	19	5	73	35	67	39
22	62	94	36	42	99	95	68	99	77	34	67	25	30	69	53
23	24	18	83	77	94	11	19	10	49	20	90	10	5	56	78
24	34	62	37	76	42	50	70	98	33	84	74	96	4	36	67
25	2	9	98	66	24	41	13	27	26	34	16	35	13	89	10
26	37	79	79	67	14	3	56	31	94	99	29	81	90	60	89
27	92	56	60	15	90	46	21	90	77	89	29	68	67	13	41
28	84	14	7	9	17	33	31	2	54	10	15	64	86	98	58
29	57	70	43	84	74	37	46	39	78	74	44	70	95	79	71
30	92	99	51	28	11	51	59	77	9	67	52	18	94	60	45
31	23	1	19	15	27	18	14	60	91	94	23	49	38	53	27
32	33	41	55	5	56	28	25	25	16	96	94	82	73	18	37
33	46	59	69	72	66	73	38	59	12	6	98	29	60	97	19
34	86	92	70	73	23	58	82	41	99	9	33	52	7	73	56
35	68	58	17	15	48	91	56	4	6	81	46	39	79	37	54
36	100	86	97	68	41	94	48	24	40	71	56	76	100	97	54
37	82	90	43	34	60	91	71	38	74	96	55	55	32	8	19
38	58	44	89	40	18	64	63	33	81	100	99	13	24	3	61
39	16	46	62	94	84	90	59	59	86	4	89	41	4	75	16

Table 4.7 Propagation Distance of transmit data between (20-39) to (16-30)

Table 4.8:

	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
20	76	23	7	77	68	72	65	42	40	82	32	82	79	86	51
21	63	3	92	81	75	82	39	62	58	54	28	25	46	23	81
22	42	61	76	59	56	59	52	9	72	100	36	98	35	89	46
23	32	18	34	22	52	91	63	11	40	6	51	44	100	82	49
24	29	24	72	63	60	67	5	35	46	25	72	86	29	74	14
25	94	40	5	35	74	80	55	69	90	6	31	5	20	73	73
26	95	55	73	58	3	45	65	53	38	94	83	85	38	60	88
27	28	72	29	90	83	40	50	70	84	61	58	33	46	72	89
28	100	56	52	34	44	50	8	89	7	44	83	40	62	82	89
29	11	39	60	46	6	23	84	2	87	8	67	51	22	58	13
30	95	66	46	84	54	56	69	37	24	58	87	41	12	45	31
31	7	44	18	3	96	44	97	77	1	69	71	65	56	22	78
32	19	1	32	70	63	55	44	29	51	77	77	58	75	65	13
33	20	35	94	40	28	16	40	38	14	44	10	62	2	58	79
34	53	83	86	79	32	46	76	11	11	27	53	98	72	32	30
35	72	88	33	66	98	8	59	42	31	27	76	100	19	79	20
36	97	12	6	31	59	54	91	55	44	55	72	2	81	15	48
37	10	47	1	92	65	1	4	21	46	13	1	73	36	79	44
38	12	41	89	55	37	21	45	96	13	48	86	5	70	98	29
39	15	61	26	33	41	41	39	61	17	19	10	33	77	24	75

Table 4.8 Propagation Distance of transmit data between (20-39) to (31-45)

Table 4.9:

	45	46	47	48	49	50
20	51	64	96	45	7	87
21	81	99	3	54	9	81
22	46	42	22	13	31	73
23	49	90	14	40	93	92
24	14	84	14	59	37	81
25	73	88	59	8	93	81
26	88	94	67	21	66	8
27	89	73	2	68	44	44
28	89	94	20	26	90	60
29	13	68	60	6	6	16
30	31	41	84	41	40	37
31	78	23	38	90	86	41
32	13	51	35	10	15	20
33	79	24	45	57	7	50
34	30	86	92	64	26	9
35	20	100	81	43	73	50
36	48	26	37	67	17	28
37	44	44	5	5	10	60
38	29	14	69	91	62	90
39	75	70	83	83	30	31

Table 4.9 Propagation Distance of transmit data between (20-39) to (45-50)

Table 4.10:

	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
40	53	6	76	61	86	99	93	41	1	55	21	22	33	10	75
41	36	5	22	40	34	23	94	69	97	44	95	1	62	81	24
42	58	85	28	63	59	97	9	51	53	10	91	89	44	79	15
43	11	4	62	57	97	75	67	53	26	97	55	4	70	52	6
44	71	88	6	22	46	96	80	46	34	6	75	51	20	43	17
45	95	53	96	8	21	78	92	79	30	16	85	79	28	23	33
46	37	31	13	92	14	34	90	50	62	59	70	3	53	4	83
47	66	30	76	56	43	27	76	90	73	41	94	26	54	96	27
48	60	29	16	1	29	56	88	5	91	14	84	81	92	14	51
49	34	84	38	83	18	13	88	5	69	74	44	38	98	40	45
50	23	22	53	44	75	8	85	68	14	86	20	61	55	17	1

Table 4.10 Propagation Distance of transmit data between (40-50) to (16-30)

Transmission No	Contention Period of Each Transmission
1	195.073736398568
2	194.264981492813
3	188.153347428521
4	188.426503058941
5	0
6	195.542832097215
7	188.080595193621
8	188.499255293841
9	0
10	203.345620873375
11	189.216958322789
12	203.313038584125
13	187.904740098220
14	188.503272288406
15	188.537639908573
16	188.507735615700
17	188.634494110863
18	188.471136331886
19	188.787139904333
20	188.118087142895
21	188.021679273335
22	188.049351902560
23	195.610228339361
24	188.315812542039
25	0
26	187.915452083727
27	188.071222206303
28	0
29	0
30	0
31	0
32	201.062182629534
33	188.013645284205
34	202.980967033418
35	188.508628281159
36	188.653686418229
37	188.347948498559
38	188.530498584902
39	0
40	202.819840918089

41	0
42	0
43	195.570951059170
44	188.125228466566
45	0

46	195.854372342368
47	0
48	202.814931258065
49	195.452226553137
50	188.131030792049
51	0
52	187.980170329496
53	0
54	196.234647827855
55	0
56	203.083623561191
57	188.635386776322
58	0
59	0
60	194.992057509080
61	188.330541522111
62	188.326078194816
63	188.644759763640
64	188.162274083110
65	0
66	0
67	0
68	0
69	0
70	0
71	0

72	194.609996692674
73	188.196195370548
74	188.349733829477
75	188.555939550480
76	203.171997441621
77	0
78	187.792710583129
79	188.196195370548
80	188.318044205686
81	0
82	203.421497437381
83	0
84	0
85	194.755501162474
86	0
87	0
88	203.317501911420
89	0
90	0
91	0
92	195.408932278381
93	0
94	203.232252360096
95	188.293049572837
96	195.234416181168
97	0
98	202.947492078710
99	188.349733829477
100	188.150223099415

Table 4.11 Contention Period for Each Transmission for CSMA/CD

For 50 nodes with 100 transmission where maximum propagation is 100nm, with few iteration,

For CSMA/CD

Total contention period for 100 transmission

12989.6429862 nanoseconds

Total collision in 100 transmissions

24

Total number of no transmission

35

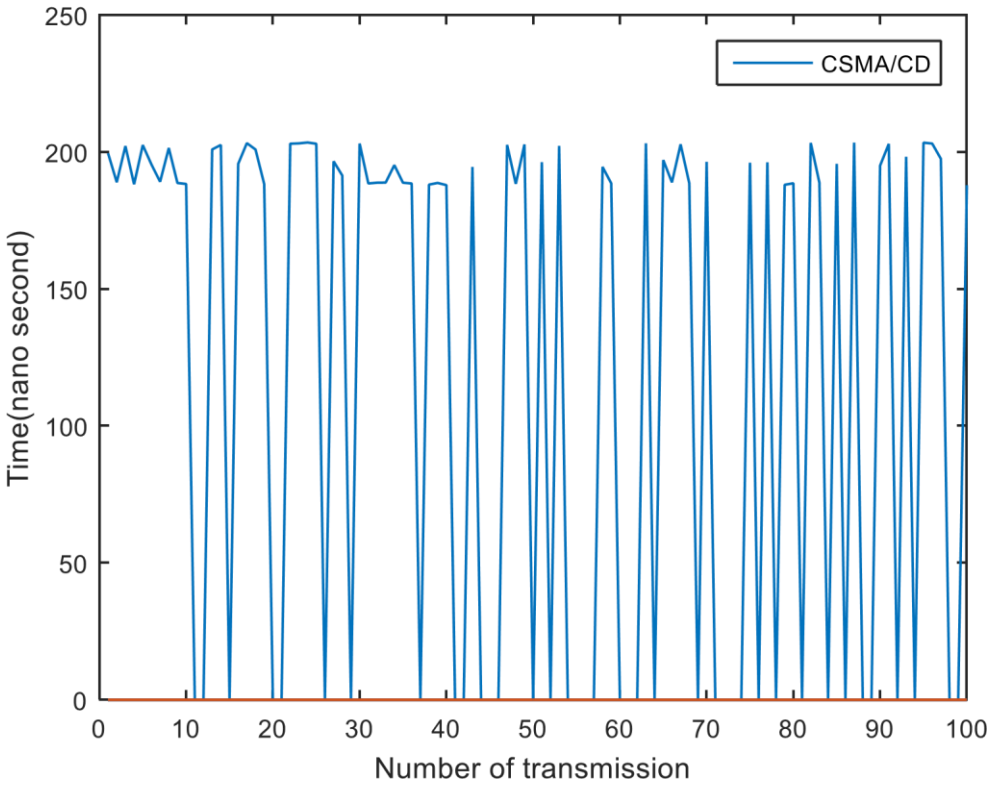

Figure 4.1 The contention period of CSMA/CD with less contention period

Here, the graph is number of node Vs contention period (nanosecond). We can see for maximum node's contention time is average 200nm. Because, the maximum propagation time in the LAN is 100. In CSMA/CD, the nodes wait for double time of maximum propagation time. In the figure 4.1, where the contention period is 0, there channel was busy so there was no transmission.

When 50 nodes want to transmit 100 data frames using proposed CSMA/CD protocol. The nodes have to wait contention period after send a data frame. Here, given contention period for every transmission. Even if, there is occurred a collision, the nodes have to wait for contention period. Only for no transmission when the channel was busy, there is no contention period.

Transmission No	Contention Period of Each Transmission
1	201.869598537100
2	13.3212466430200
3	30.7951730007976
4	93.5517864244330
5	94.3864286284954
6	125.805574785057
7	157.137686059377
8	125.384236688461
9	79.0772160085268
10	110.144651974286
11	0.561932906371444
12	149.160827518734
13	38.6403633862550
14	0
15	39.5191925305325
16	171.194935373252
17	187.702105039052
18	0
19	172.230873638295
20	110.089306715835
21	125.454757259713
22	187.959192691212
23	63.0480686959631
24	110.467350537675
25	125.770314499431
26	125.815394105105
27	79.0209780846167
28	0
29	0
30	116.841428246881
31	179.871643633666
32	85.8717391488703
33	0
34	0
35	155.949102000865
36	16.4611973946666

37	79.0370460628767
38	0
39	0
40	195.563809735499
41	0
42	0
43	23.4730845742499
44	0
45	0
46	108.271393508805
47	172.395124082730
48	0
49	8.09335138302891
50	93.4643052094617
51	0

52	38.8081844925264
53	188.464887673674
54	0
55	0
56	155.816094847490
57	0
58	133.007599707384
59	110.174556267159
60	0
61	70.6303691037773
62	0
63	117.385507844075
64	0
65	0
66	38.2431272570488
67	0
68	0

69	0
70	0
71	0
72	0
73	121.818930845653
74	0
75	0
76	0
77	0
78	121.901948733330
79	0
80	0
81	54.9029427163185
82	125.089657087027
83	0
84	155.754054598097
85	0
86	0
87	194.456904566475
88	0
89	0
90	0
91	178.627267983973
92	0
93	86.1288268010307
94	15.6636008071481
95	30.7474153987470
96	157.071182482690
97	109.711709226724
98	203.274207636664
99	63.0436053686686

100	0

Table 4.12 Contention Period for Each Transmission for Proposed CSMA/CD

For 50 nodes with 100 transmission where maximum propagation is 100nm, with few iteration,

Proposed Enhanced CSMA/CD

Total Contention period for 100 transmissions

6178.343327900 nanoseconds

Total collision in 100 transmissions

25

Total number of no transmission

39

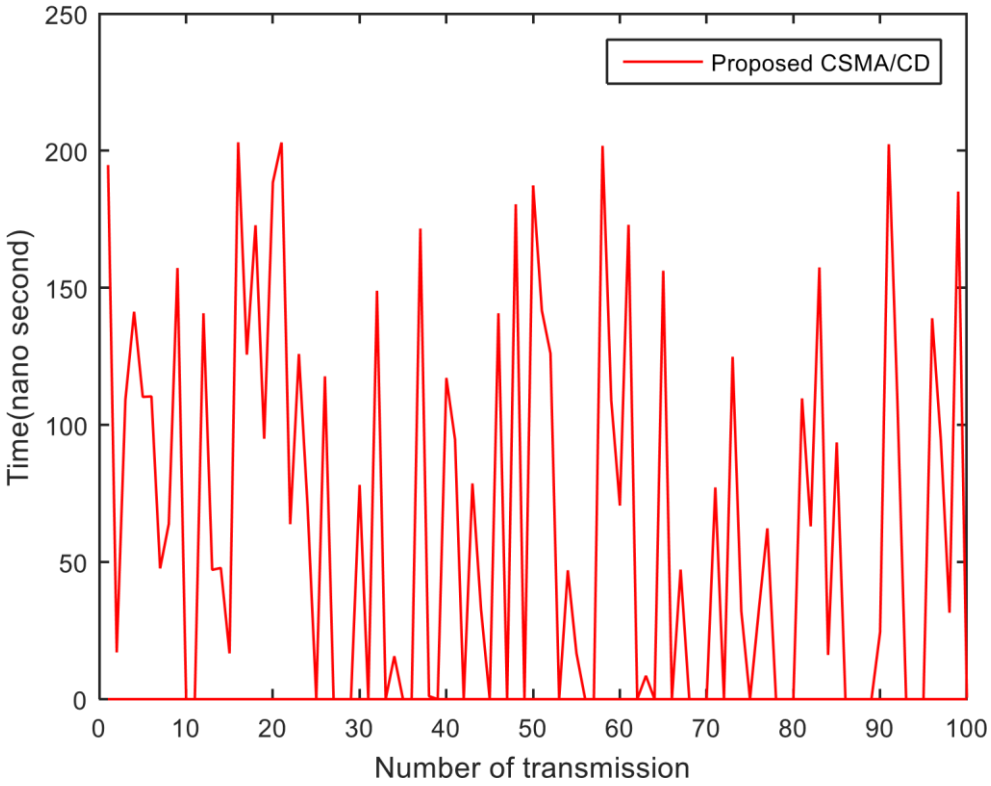

Figure 4.2The contention period of proposed CSMA/CD

Here, the graph is number of node Vs contention period (nanosecond). In this figure 4.2, variety in contention time has been shown. But the contention period are not more 200nm. Because, the

maximum propagation time in the LAN is 100. In CSMA/CD, the nodes wait for double time of maximum propagation time, but we use the double of propagation time between sender and receiver as contention period. In the figure 4.2, where the contention period is 0, there channel was busy so there was no transmission.

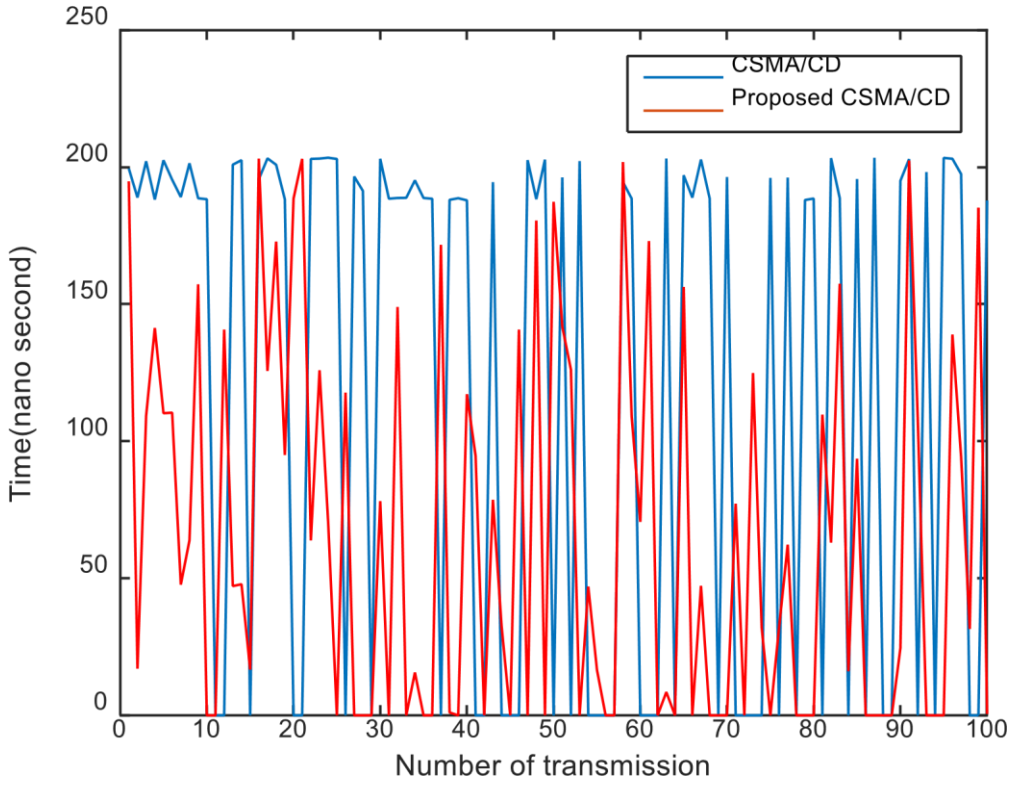

Figure 4.3 The Combined Graph of Contention Period of CSMA/CD and CSMA/CD update

Here, in the figure 4.3, the compare between the contention period of CSMA/CD and proposed CSMA/CD (which method we use in this thesis paper).

The total time in the proposed CSMA/CD of contention period is less than CSMA/CD.

Discussion

The simulation shows the compare between contention period of CSMA/CD and proposed CSMA/CD. The total contention time of CSMA/CD is higher than proposed CSMA/CD. The node have to wait for a long time in CSMA/CD rather than proposed CSMA/CD. In figure 4.3, the blue line indicates the contention period for CSMA/CD and the red line indicates the contention period for proposed CSMA/CD. From the figure 4.3, when the number of transmission is 10, then the contention period of CSMA/CD is about 200nanoseconds, but the contention period of proposed CSMA/CD is about 110nanoseconds, which is almost half of contention period of CSMA/CD. Thus, proposed CSMA/CD reduce the contention period of CSMA/CD.

Chapter 5

Conclusion

We have successfully implemented our method on reducing contention period of CSMA/CD. The data is created for an artificial LAN, which has some nodes and channels. The nodes can transmit data frame easily to another node when the both nodes and channel are free. But when one of the node is busy with transmitting data with another node, the sender can not sent data to the receiver node. If the both nodes are free, the sender and receiver node have to wait for long time before according to CSMA/CD algorithm. Whereas now, using our proposed method the nodes have to wait less time than before. In this thesis paper, we explain the CSMA/CD method properly with it's background and try to prove that our propose method works better than CSMA/CD.

We have not simulated the proposed method in real life. But we simulate it theoretically using MATLAB R2015a. We hope that our method works in real life as well as theoretically. Moreover, we use short data for simulation. Our method is also useful for a large LAN, where we considered $3m^2$ area LAN. Generally, area of a LAN is $100m^2$, our proposed method will give better result than CSMA/CD in a large LAN. According to CSMA/CD, the total contention period of a LAN is long time where a number of nodes want to transmit data to another node. Furthermore, we compare with our new method with the CSMA/CD and prove that our method is successfully reduced the contention period than the contention period of CSMA/CD.

This proposed method can be much better if the method can be use in real life, and there occurs a problem in real life use, then it can be try to solve out in future. If we use the proposed method in real life, it will useful for transmission among the nodes.

Reference

[1] **“**The Evolution of Ethernet” [Online]. Available:

https://www.safaribooksonline.com/library/view/ethernet definitive/1565926609/ch01.html

[2] “OSI reference model (Open Systems Interconnection)”[Online].

Available: http://searchnetworking.techtarget.com/definition/OSI

[3] “Computer Networks (CS425) ISO-OSI 7-Layer Network Architecture”[Online].

Available:https://www.cse.iitk.ac.in/users/dheeraj/cs425/lec01.html

[4] “Medium access control”[Online].

Available: https://en.wikipedia.org/wiki/Medium_access_control

[5] “MAC Algorithms in Wireless Networks”[Online].

Available:https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0ahUKEwj8hpmZe3XAhUHwI8KHWagA4sQFgglMAA&url=http%3A%2F%2Fciteseerx.ist.psu.edu%2Fviewdoc%2Fdownload%3Fdoi%3D10.1.1.65.3965%26rep%3Drep1%26type%3Dpdf&usg=AOvVaw0d8rIlBLJKLfZ_qm-f0vgP

[6] “Carrier-sense multiple access”[Online].

Available: https://en.wikipedia.org/wiki/Carrier-sense_multiple_access

[7] “Carrier Sense Multiple Access with Collision Detection (CSMA/CD)”[Online].

Available: http://ecomputernotes.com/computernetworkingnotes/multiple-access/ carrier-sense-multiple-access-with-collision-detection

[8] “Contention”[Online] Available: https://www.webopedia.com/TERM/C/contention.html

[9] "MAC Algorithms in Wireless Networks"[Online].

Available:https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0ahUKEwj8hpmZe3XAhUHwI8KHWagA4sQFgglMAA&url=http%3A%2F%2Fciteseerx.ist.psu.edu%2Fviewdoc%2Fdownload%3Fdoi%3D10.1.1.65.3965%26rep%3Drep1%26type%3Dpdf&usg=AOvVaw0d8rIlBLJKLfZ_qm-f0vgP

[10] "Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)"[Online].

Available access/carrier-sense-multiple-access-with-collision-avoidance

[11] "CSMA/CN: Carrier Sense Multiple Access with Collision Notification"[Online].

Available: https://pdfs.semanticscholar.org/3b41/3807289977602991ef26de2b8dd84337d93c.pdf

[12] Marie Duflot, Laurent Fribourg, "Probabilistic Model Checking of the CSMA/CD protocol using PRISM and APMC"

Available:http://www.prismmodelchecker.org/papers/avocs04.pdf

[13] **"Carrier Sensed Multiple Access (CSMA)"[Online].**

Available:http://ecomputernotes.com/computernetworkingnotes/multiple-access/what-is-csma-difference-netween-csmaca-and-csmacd

[14] Nasir A. Malik, "PERFORMANCE EVALUATION OF CSMA/CD LOCAL AREA NETWORK"Available:https://ttu-ir.tdl.org/ttu-ir/bitstream/handle/2346/16214/31295004922349.pdf;jsessionid=FD878B2821722E856E30CB98924B ABD?sequence=1

[15] Abbey Wineland, "Modifying the CSMA/CD (IEEE 802.3) Protocol to Involve a Network Queue (CSMA/CDNQ)" Available:

https://pdfs.semanticscholar.org/9086/2c6311a984e8cd05d3a026da923adcbf0433.pdf

Appendix

Essential Source Code Of This Thesis Paper

----------**MatLab Programming** ----------

CSMA/CD:

```
%This is CSMA/CD Protocol z=

zeros(100); % time

%data table for distance time for each node  x=zeros(100);

status= zeros(50,50); %Channel Status (free or busy) r=0;

p=0;  c=0; for data1=1:50 for data2=1:50     q=randi(100);

n(data1,data2)=q; %data table for distance time for each node

end end
```

```
%n=xlsread('mydata.xlsx'); %Load the distance time for each node from excel file.
%tic; fori=1:100 % this loop is for 100 transmission for m=1:50 for k=1:50
m1=randi(2); status(i,k)=m1; %randomly update the channel status end end
a=randi(50); %Randomly generate sender     b=randi(50); %Randomly
generate the receiver     c=2*100*.001; %Here calculate the waiting time
for sender and receiver.
%Maximum Propagation time is 100seconds.
%The time is calculated in nano seconds.                    %n(a,b)
is the distance between sender and
%receiver.
if status(a,b)==1 && status(a,b)==1 %when both node are free
tic;
pause(c); %waiting for twice of distance time between the node
d=toc; z(i)=d*1000; %calculate the propagation time for
each node
%there is no collision.
elseif status(a,b)==2 && status(a,b)==2 %when both node are busy
z(i)=0;
%There is no transmission else
%there occurs a collision tic;
pause(c); %waiting for twice of distance time between the node
```

d=toc; z(i)=d*1000; **%calculate the propagation time for**

each node

%number of collision p=p+1; end r=r+z(i);

%total Propagation time x(i)=i; end disp('Total

propagation time for 100 transmission'); disp(r);

disp('Total collision in 100 transmission'); disp(p);

Proposed Enhanced CSMA/CD

%This is CSMA/CD Protocol y=

zeros(100); **%propagation time**

%n= zeros(50,50); **%data table for**

distance time for each node x=

zeros(100); status= zeros(50,50);

%Channel Status (free or busy)

r=0; p=0; c=0; for data1=1:50 for

data2=1:50 q=randi(100);

n(data1,data2)=q;**%data table for**

distance time for each node end

end

%n=xlsread('mydata.xlsx'); **%load the distance time for each node from excel file**.

%tic; fori=1:100 **% this loop is for 100**

transmission for m=1:50 for k=1:50

m1=randi(2);

status(i,k)=m1; **%randomly update the channel status**

end end a=randi(50); **%randomly generate sender**

b=randi(50); **%randomly generate the receiver**

c=2*n(a,b)*0.001; **%here calculate the waiting time**

for sender and receiver.

%the time is calculated in nano second.

%n(a,b) is the distance between sender and %receiver.

if status(a,b)==1 && status(a,b)==1 **%when both node are free**

tic;

pause(c); **%waiting for twice of distance time between the node**

d=toc; y(i)=d*1000; **%calculate the propagation time for**

each node

%there is no collision.

elseif status(a,b)==2 && status(a,b)==2 **%when both node are busy** y(i)=0;

%there is no transmission

else

%there occurs a collision

tic;

pause(c); **%waiting for twice of distance time between the node**

d=toc; y(i)=d*1000; **%calculate the propagation time for each node** **%number of collision** p=p+1; end r=r+y(i); **%total Propagation time** x(i)=i; end disp('Total propagation time for 100 transmission'); disp(r); disp('Total collision in 100 transmission'); disp(p); plot(x,z); holdon;plot(x,y);hold off; xlabel('Number of transection'); ylabel('Time(nano second)'); %plot(x,y); legend('CSMA/CD', 'CSMA/CD UP');

C++ Code:

Main.cpp

```
#include <iostream>
#include "Station.h"
#include <time.h>
#include <stdlib.h>
#include <windows.h>
using namespace std; bool
randomBool() {     return
rand() % 2 == 1;
}
int main()
{
```

```
Station *s1, *s2, *s3;     float r1, r2, r3, r4, r5;     bool carrierStatus;

for(inti=0; i<500; i++){         for(int j=0; j<3; j++){             carrierStatus =

randomBool();             r1 = static_cast<float> (rand()) / static_cast<float>

(RAND_MAX);             s1 = new Station(r1, carrierStatus);

carrierStatus = randomBool();             r1 = static_cast<float> (rand()) /

static_cast<float> (RAND_MAX);             s2 = new Station(r1,

carrierStatus);             carrierStatus = randomBool();             r1 =

static_cast<float> (rand()) / static_cast<float> (RAND_MAX);             s3 =

new Station(r1, carrierStatus);             cout<<"Sending message form s1 to

s2"<<endl<<endl;             if(s1->getCarrierStatus() && s2-

>getCarrierStatus()){                 cout<<"Collision occur! Transmiss

stop!"<<endl<<endl;                 r4 = static_cast<float> (rand()) /

static_cast<float> (RAND_MAX);

Sleep(r4);

}

else if(s1->getCarrierStatus()==false && s2->getCarrierStatus() == false){

r2 = static_cast<float> (rand()) / static_cast<float> (RAND_MAX);                 s2-

>setMessage(r2);                 cout<<"Data has been transmited

successfully!"<<endl<<endl;

}             else{

cout<<"s1 or s2 is looking busy!"<<endl<<endl;                 r3 =

static_cast<float> (rand()) / static_cast<float> (RAND_MAX);
```

```
Sleep(r3);
}
cout<<"Sending message form s2 to s3"<<endl<<endl;            if(s2-
>getCarrierStatus() && s3->getCarrierStatus()){ cout<<"Collision occur!
Transmiss stop!"<<endl<<endl; r4 = static_cast<float> (rand()) /
static_cast<float> (RAND_MAX);
Sleep(r4);
}
else if(s2->getCarrierStatus()==false && s3->getCarrierStatus() == false){
r2 = static_cast<float> (rand()) / static_cast<float> (RAND_MAX);          s3-
>setMessage(r2);                cout<<"Data has been transmited
successfully!"<<endl<<endl;
}            else{
cout<<"s2 or s3 is looking busy!"<<endl<<endl;                r3 =
static_cast<float> (rand()) / static_cast<float> (RAND_MAX);
Sleep(r3);
}

cout<<"Sending message form s3 to s1"<<endl<<endl;            if(s3-
>getCarrierStatus() && s1->getCarrierStatus()){                cout<<"Collision
occur! Transmiss stop!"<<endl<<endl;                r4 = static_cast<float>
(rand()) / static_cast<float> (RAND_MAX);
```

```
Sleep(r4);
}
else if(s3->getCarrierStatus()==false && s1->getCarrierStatus() == false){ r2 =
        static_cast<float> (rand()) / static_cast<float> (RAND_MAX); s1-
        >setMessage(r2);
cout<<"Data has been transmited successfully!"<<endl<<endl;
}       else{
cout<<"s3 or s1 is looking busy!"<<endl<<endl;          r3 =
static_cast<float> (rand()) / static_cast<float> (RAND_MAX);
Sleep(r3);
}

cout<<"Sending message form s3 to s2"<<endl<<endl;      if(s3-
>getCarrierStatus() && s2->getCarrierStatus()){          cout<<"Collision
occur! Transmiss stop!"<<endl<<endl;          r4 = static_cast<float>
(rand()) / static_cast<float> (RAND_MAX);
Sleep(r4);
}
else if(s3->getCarrierStatus()==false && s2->getCarrierStatus() == false){
r2 = static_cast<float> (rand()) / static_cast<float> (RAND_MAX);          s2-
>setMessage(r2);          cout<<"Data has been transmited
successfully!"<<endl<<endl;
```

```
}           else{
             cout<<"s3 or s2 is looking busy!"<<endl<<endl; r3 =
             static_cast<float> (rand()) / static_cast<float> (RAND_MAX);
             Sleep(r3);

}
}
}
}
```

Station.cpp

```
#include "Station.h"
Station :: Station(int message, int status){
_message = message;
_carrierStatus = status;
}

int Station::getMessage(){
return _message;
}
```

```
bool Station::getCarrierStatus(){

return _carrierStatus;

}


int Station::getReceivedMessage(){     return _receivedMessage;

}


void Station::setMessage(int message){

_message=message;

}


void Station::setCarrierStatus(bool status){

_carrierStatus = status;

}


void Station::setReceivedMessage(int message){

_receivedMessage = message;

}
```

Station.h

```
using namespace std;
```

```
class Station{

private:

int _message;        bool

_carrierStatus;        int

_receivedMessage;

public:


Station(int message, intcarrierStatus);

intgetMessage();        bool getCarrierStatus();

intgetReceivedMessage();        void setMessage(int

message);        void setCarrierStatus(bool status);

void setReceivedMessage(intreceivedMessage);

};
```

57

58