NITTE EDUCATION TRUST | NMAM INSTITUTE OF TECHNOLOGY

A Project Report on

Performance evaluation of TDMA vs CSMA-CA

Project Associates

USN		Name
4NM19EC055		Glenn Elish Peter
4NM19EC056		Gowtham
4NM19EC057		Guruprasad Nayak
4NM19EC058		H Pranav Nayak

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

N.M.A.M. INSTITUTE OF TECHNOLOGY, NITTE - 574110

December 2022

NITTE EDUCATION TRUST | NMAM INSTITUTE OF TECHNOLOGY

Department of Electronics and Communication Engineering

CERTIFICATE

This is to certify that Glenn Elish Peter (4NM19EC055), Gowtham (4NM19EC056), Guruprasad Nayak (4NM19EC057) and H Pranav Nayak (4NM19EC058) are Bonafede students of N.M.A.M. Institute of Technology, Nitte have submitted the report for the project entitled "Performance evaluation of TDMA vs CSMA-CA" towards Computer Networks Lab (19EC702) in partial fulfilment of the requirements for the award of Bachelor of Engineering Degree in Electronics and Communication Engineering during the year 2022-2023.

Name of the Examiner	Signature with date
1. Prajwal Hegde N	27/12/22
2. Kavitha S	27/12/22

ABSTRACT

Time Division Multiple Access (TDMA) is a probabilistic Media Access Control (MAC) protocol which is a channel access method for shared medium networks. It allows several users to share the same frequency channel by dividing the signal into different time slots. Carrier Sense Multiple Access (CSMA) is also a probabilistic MAC in which a node verifies the absence of other traffic before transmitting on a shared transmission medium. This project has been implemented using ns2 to simulate the TDMA to transmit data at first, and then simulate the data transfer with CSMA/CA. After the simulations, AWK is used to analyze data and then X-graph to plot the end-to-end delay with two different amount packet sizes 48 and 4800 for two protocols. After the simulation, TDMA has a larger delay time than CSMA/CA, but TDMA delay is more stable. Hence TDMA is considerably a better protocol then CAMA/CA.

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

Time Division Multiple Access (TDMA), a second-generation (2G) technology used in digital cellular telephone communication, is a probabilistic Media Access Control (MAC) protocol in which is a channel access method for shared medium networks. It is a method that divides the spectrum into time slots so that more amounts of data can be carried. In this way, a specific frequency is not one-to -one to a user, but a single frequency can provide multiple data channels to co-response multiple users. Although TDMA is considered the most unadvanced second-generation technology, it was widely used all over the world in the past few years. The statistics show that about 9% of digital cell phone users chose TDMA in the US in 1999. Like TDMA, CSMA is also a MAC. It can detect the absence of other traffic before transmitting on a shared transmission medium. There are two modifications of CSMA, the one is called CSMA/CD, and the other one is CSMA/CA which is used in the project, where CD is short for collision detection and CA stands for collision avoidance. CSMA/CD refers to using the terminating transmission to detect and deal with the collision and then prevent the collision happening again to improve the performance. On the other hand, CSMA/CA acts to reduce the probability of the first-time collision happening on the channel. Because it checks if the channel is clear or not once a node receives a packet. If the channel is idle, then the packet is sent; however, if the channel is busy, the node will wait a period of time and then check it again until the channel is clear. That ensures there is only one node transmitting at one time so that prevents the collision fundamentally. According to the end-to-end delay, the performance with different amounts of packet size has been evaluated. During simulation, two setting points of packet size are considered which are 48 and 4800. In the end, the comparison and analysis of the result has been performed.

1.1 Literature Review

To make this project TDMA and CSMA were studied briefly and understood the need and complexities of both the methods. Many papers and pre-requisites regarding TDMA and CSMA are studied. Some papers explained the working of TDMA and some other papers explained about CSMA and its methods.

There is a great amount of research related to TDMA and CSMA at present. TDMA is a 2G technology used in digital cell telephone communication which uses probabilistic MAC protocol which is a channel access method for shared medium network. It allows several users to share the same frequency channel by dividing the signal into different time slots [3].

CSMA is also a probabilistic MAC in which a node verifies the absence of other traffic before transmitting on a shared transmission medium. It is a method that can divide the spectrum into time slots so that more amounts of data can be carried. In this way, a specific frequency is not one-to -one to a user, but a single frequency can provide multiple data channels to co response multiple users. Although TDMA is considered the most unadvanced second-generation technology, it was widely used all over the world in the past few years [4].

TDMA and CSMA are probabilistic media Access Control that can detect the absence of other traffic before transmitting on a shared transmission medium.

There are two modifications of CSMA, the one is called CSMA/CD, and the other one is CSMA/CA. CSMA/CA reduces the probability of the first-time collision happening on the channel, because it checks if the channel is clear or not once a node receives a packet. If the channel is idle, then the packet is sent; however, if the channel is busy, the node will wait for a period of time and then check it again until the channel is clear. That ensures there is only one node transmitting at one time so that prevents the collision fundamentally.

1.2 Objectives

1. Using “Network Simulator-2” to simulate the data transfer in TDMA protocol and CSMA protocol, and compare their delay time to analyze their performances.
2. Writing .tcl files for TDMA protocol and CSMA protocol in NS2, and running the simulation.
3. Using the delay.awk to filter out all useful data and save into tdma.txt and csma.txt files.
4. Using X-graph to plot the end-to-end delay of two different protocols in the same graph.

Chapter 2

METHODOLOGY

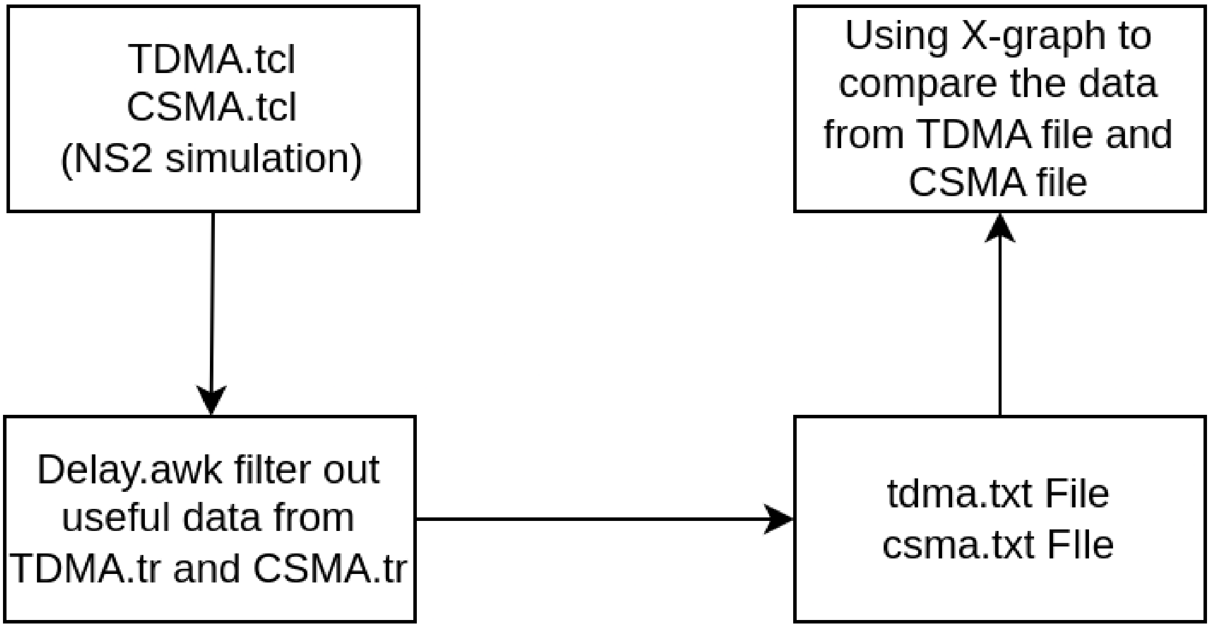

Figure 1. Flow chart of the project

Set some initial configurations such as channel type, network interface type, etc are set. The values from val(chan) to val(nn) are for nodes. val(x), val(y), and val(stop) are for ns2.

note: mac/802_11 is csma; mac/tdma is tdma

- Set up a new Simulator; create TDMA.tr and TDMA.nam or CSMA.tr and CSMA.nam in order to record the data during the simulation.
- Build topography, and sign a general operations director (god) which can manage nodes and make sure all nodes are in the area. Configure the nodes, but only turn on MAC traces; also create 4 nodes.
- Provide the initial location of nodes, so that it is easier to monitor. Assign the node0 is a sender, the node1 is a receiver, and node2, 3 are transporters, where X, Y, Z value represents the coordinate. Since it is a 2D graph, all the Z values are 0. For example, node 0 is located at 5.0, 5.0.

- Set a TCP connection between node0 and node1. Attach the CBR into the TCP, define some parameters; CBR starts sending data every second after 10s later.
- AWK filters out the useful data from TDMA.tr or CSMA.tr. In TDMA.tcl and CSMA.tcl, the record of all data in the Mac layer will be stored in TDMA.tr and CSMA.tr. The interest is only when node0 sends the packet to node1, and when node1 receives the same packet. The time about sending and receiving is collected and sending time is subtracted with receiving time in order to get the delay time for one signal packet. If all the delay time for every transported packet has been achieved, then this data is saved into TDMA and CSMA.
- After the AWK filters out the useful data and stores it into 2 files, X-graph software is used to plot the end-to-end graph, where X axis is the simulating time; Y axis is the delay time and red curve represents TDMA and blue curve stands for CSMA.

2.1 Code

2.1.1 tdma.tcl

```
# Define options
set val(chan) Channel/WirelessChannel ;# channel type
set val(prop) Propagation/TwoRayGround ;# radio-propagation model
set val(netif) Phy/WirelessPhy ;# network interface type
set val(mac) Mac/Tdma ;# MAC type
set val(ifq) Queue/DropTail/PriQueue ;# interface queue type
set val(ll) LL ;# link layer type
set val(ant) Antenna/OmniAntenna ;# antenna model
set val(ifqlen) 50 ;# max packet in ifq
set val(nn) 4 ;# number of mobilenodes
set val(rp) AODV ;# routing protocol
set val(x) 500 ;# X dimension of topography
set val(y) 400 ;# Y dimension of topography
set val(stop) 150 ;# time of simulation end
```

```
set ns [new Simulator]
set tracefd [open tdma.tr w]
set namtrace [open tdma.nam w]
$ns trace-all $tracefd
$ns namtrace-all-wireless $namtrace $val(x) $val(y)
# set up topography object
set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)
create-god $val(nn)
#
# Create nn mobilenodes [$val(nn)] and attach them to the channel.
#
# configure the nodes
$ns node-config -adhocRouting $val(rp) \
-llType $val(ll) \
-macType $val(mac) \
-ifqType $val(ifq) \
-ifqLen $val(ifqlen) \
-antType $val(ant) \
-propType $val(prop) \
-phyType $val(netif) \
-channelType $val(chan) \
```

```
-topoInstance $topo \
-agentTrace OFF \
-routerTrace OFF \
-macTrace ON \
-movementTrace OFF

for {set i 0} {$i < $val(nn) } { incr i } {
set node_($i) [$ns node]
}

# Provide initial location of mobilenodes
$node_(0) set X_ 5.0
$node_(0) set Y_ 5.0
$node_(0) set Z_ 0.0

$node_(1) set X_ 490.0
$node_(1) set Y_ 285.0
$node_(1) set Z_ 0.0

$node_(2) set X_ 100.0
$node_(2) set Y_ 70.0
$node_(2) set Z_ 0.0

$node_(3) set X_ 250.0
$node_(3) set Y_ 240.0
$node_(3) set Z_ 0.0

# Set a TCP connection between node_(0) and node_(1)
set tcp [new Agent/TCP/Newreno]
$tcp set class_ 2
```

```
set sink [new Agent/TCPSink]
$ns attach-agent $node_(0) $tcp
$ns attach-agent $node_(1) $sink
$ns connect $tcp $sink
set e [new Application/Traffic/CBR]
$e attach-agent $tcp
$e set packetSize_ 4800
$e set rate_ 6kb
$e set interval_ 1
$ns at 10.0 "$e start"
# Define node initial position in nam
for {set i 0} {$i < $val(nn)} { incr i } {
# 30 defines the node size for nam
$ns initial_node_pos $node_($i) 30
}
# Telling nodes when the simulation ends
for {set i 0} {$i < $val(nn) } { incr i } {
$ns at $val(stop) "$node_($i) reset";
}
# ending nam and the simulation
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "stop"
$ns at 150.01 "puts \"end simulation\" ; $ns halt"
proc stop {} {
global ns tracefd namtrace
$ns flush-trace
```

```
close $tracefd
close $namtrace
#Execute nam on the trace file
exec nam tdma.nam &
exec awk -f delay.awk tdma.tr > tdma.txt &
exit 0
}

#Call the finish procedure after 5 seconds of simulation time
$ns run
```

2.1.2 csma.tcl

```
# Define options
set val(chan) Channel/WirelessChannel ;# channel type
set val(prop) Propagation/TwoRayGround ;# radio-propagation model
set val(netif) Phy/WirelessPhy ;# network interface type
set val(mac) Mac/802_11 ;# MAC type
set val(ifq) Queue/DropTail/PriQueue ;# interface queue type
set val(ll) LL ;# link layer type
set val(ant) Antenna/OmniAntenna ;# antenna model
set val(ifqlen) 50 ;# max packet in ifq
set val(nn) 4 ;# number of mobilenodes
set val(rp) AODV ;# routing protocol
set val(x) 500 ;# X dimension of topography
set val(y) 400 ;# Y dimension of topography
set val(stop) 150 ;# time of simulation end

set ns [new Simulator]
set tracefd [open csma.tr w]
```

```
set namtrace [open csma.nam w]
$ns trace-all $tracefd
$ns namtrace-all-wireless $namtrace $val(x) $val(y)
# set up topography object
set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)
create-god $val(nn)
#
# Create nn mobilenodes [$val(nn)] and attach them to the channel.
#
# configure the nodes
$ns node-config -adhocRouting $val(rp) \
-llType $val(ll) \
-macType $val(mac) \
-ifqType $val(ifq) \
-ifqLen $val(ifqlen) \
-antType $val(ant) \
-propType $val(prop) \
-phyType $val(netif) \
-channelType $val(chan) \
-topoInstance $topo \
-agentTrace OFF \
-routerTrace OFF \
```

```
-macTrace ON \
-movementTrace OFF

for {set i 0} {$i < $val(nn) } { incr i } {
set node_($i) [$ns node]
}

# Provide initial location of mobilenodes
$node_(0) set X_ 5.0
$node_(0) set Y_ 5.0
$node_(0) set Z_ 0.0

$node_(1) set X_ 490.0
$node_(1) set Y_ 285.0
$node_(1) set Z_ 0.0

$node_(2) set X_ 100.0
$node_(2) set Y_ 70.0
$node_(2) set Z_ 0.0

$node_(3) set X_ 250.0
$node_(3) set Y_ 240.0
$node_(3) set Z_ 0.0

# Set a TCP connection between node_(0) and node_(1)
set tcp [new Agent/TCP/Newreno]
$tcp set class_ 2
set sink [new Agent/TCPSink]
$ns attach-agent $node_(0) $tcp
$ns attach-agent $node_(1) $sink
```

```
$ns connect $tcp $sink
set e [new Application/Traffic/CBR]
$e attach-agent $tcp
$e set packetSize_ 4800
$e set rate_ 6kb
$e set interval_ 1
$ns at 10.0 "$e start"
# Define node initial position in nam
for {set i 0} {$i < $val(nn)} { incr i } {
# 30 defines the node size for nam
$ns initial_node_pos $node_($i) 30
}
# Telling nodes when the simulation ends
for {set i 0} {$i < $val(nn) } { incr i } {
$ns at $val(stop) "$node_($i) reset";
}
# ending nam and the simulation
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "stop"
$ns at 150.01 "puts \"end simulation\" ; $ns halt"
proc stop {} {
global ns tracefd namtrace
$ns flush-trace
close $tracefd
close $namtrace
#Execute nam on the trace file
exec nam csma.nam &
```

108. exec awk -f delay.awk csma.tr > csma.txt &

109. exit 0

110. }

111.

112. #Call the finish procedure after 5 seconds of simulation time

113. $ns run

2.1.3 delay.awk

1. #Initial the program, and set max packetID# to 0

2. BEGIN {

3. highest_uid = 0;

4.

5. }

6.

7. #Define every column of the tr file

8.

9. {

10.

11. event = $1; #first column means that actions such as receiving and sendin.

12.

13. time = $2; #time

14.

15. node_nb = $3; #node number

16.

17. node_nb=substr(node_nb,2,1); #in tr file, node number is like _3_, but we only need 3, so we remove '_'

18.

19. trace_type = $4; #trac_type such as Mac

20.

```
flag = $5; #
uid = $6; #packetID
pkt_type = $7; #Packet type
pkt_size = $8; #packet size
# Record the max CBR packet ID in Mac, and assign this ID to highest _uid
if ( event=="s" && node_nb==0 && pkt_type=="tcp" && uid > highest_uid && trace_type=="MAC")
{
highest_uid = uid;
}
#record the packet sending time and save into start_time array
if ( event=="s" && node_nb==0 && pkt_type=="tcp" && uid==highest_uid&&trace_type=="MAC" && pkt_size="4800")
{
start_time[uid] = time;
}
#record the packet receiving time and save into end_time array
if ( event=="r" && node_nb ==1 && pkt_type=="tcp" && uid==highest_uid&& trace_type=="MAC" && pkt_size="4800")
end_time[uid] = time;
```

```
}
END {
# calculate the delay time
for ( packet_id = 0; packet_id <= highest_uid; packet_id++ )
{
start = start_time[packet_id];
end = end_time[packet_id];
packet_duration = end - start;
#print out the positive delay time
if ( start <end )
printf("%d %f\n", packet_id, packet_duration);
}
}
```

Chapter 3

RESULTS

From the simulation result, it is found that if network data volume is within the network can withstand, TDMA will waste a lot of slot time. On the other hand, in the same situation, the end-to-end delay of CSMA is smaller than that of TDMA. Comparing Figure 3 and Figure 4, with a larger throughput, the delay of TDMA network transmission tends to stably, but the delay of CSMA transmission is getting larger and unstable. Even though from the graphs, the delay of CSMA is still smaller than the delay of TDMA, we can expect easily that with the increasing of the throughput, the delay of CSMA will be larger than the delay of TDMA and because of the instability of CSMA, it may be disconnected in a very large throughput.

Today, a huge amount of data transfer is required. Even though CSMA has a smaller delay, TDMA is much more stable. Therefore, TDMA is considered as a better protocol in two test cases. In this project, only two packet sizes are simulated. To make the simulation more accurate, more packet sizes can be simulated and a new graph, delay vs. size of packet graph, can be drawn. To make a better protocol, TDMA and CSMA can be combined into a new protocol. In a small amount of data, the new protocol performs as CSMA and in a huge amount of data, the new protocol performs as TDMA.

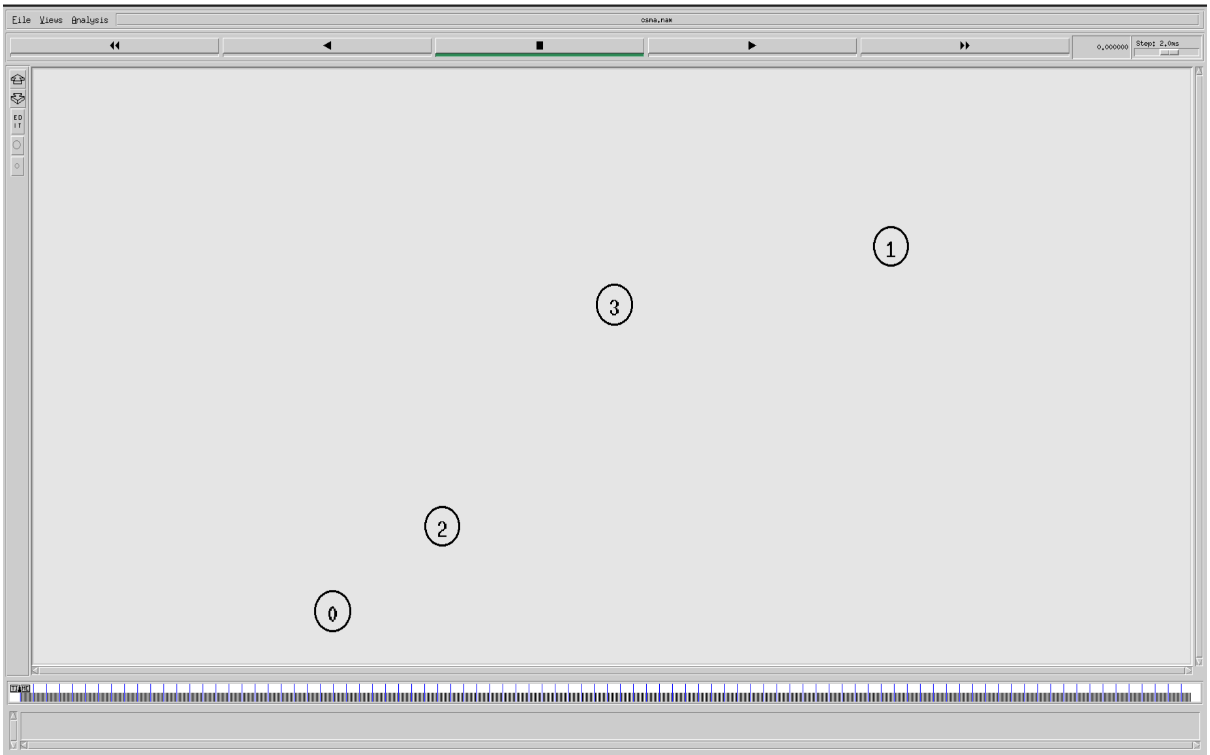

FIgure 2. nam scenario

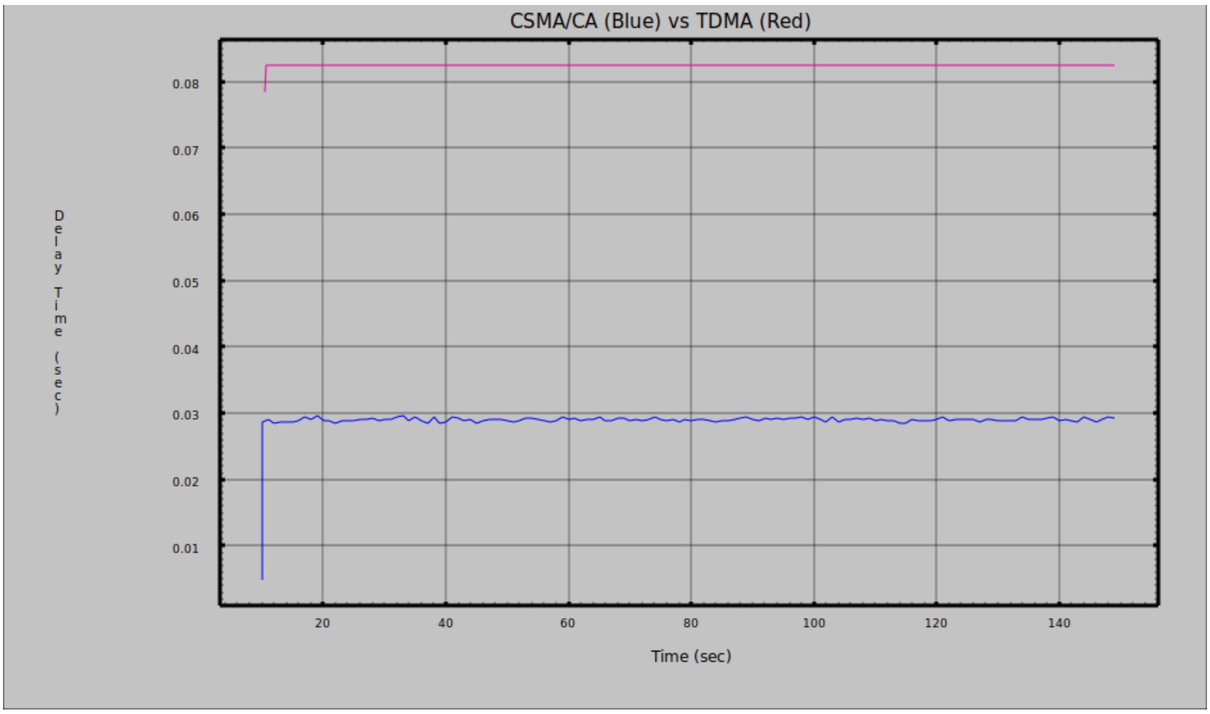

FIgure 3. CSMA/CA vs TDMA with packet size equal to 48

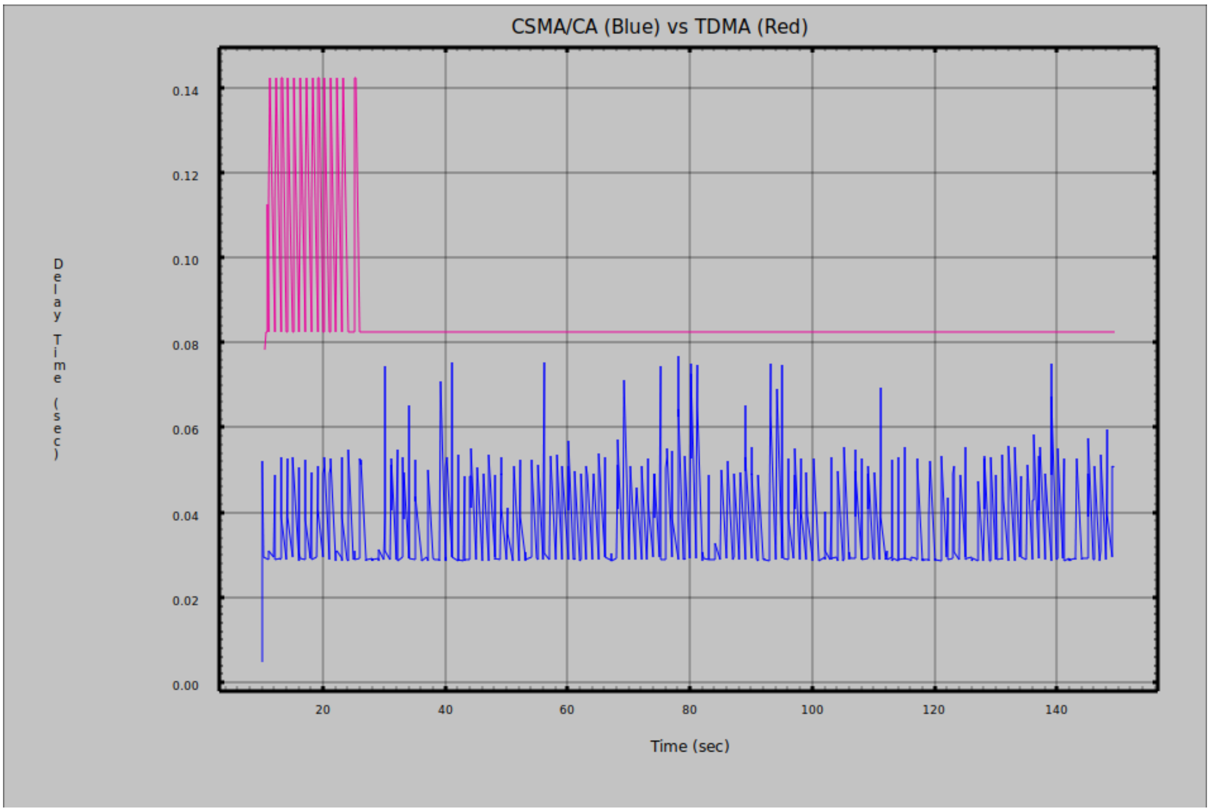

FIgure 4. CSMA/CA vs TDMA with packet size equal to 4800

REFERENCES

[1] NS2 manual [online]
https://networksimulator2.com/ns2-manual/

[2] Reading trace formats [online]
https://nsnam.sourceforge.net/wiki/index.php/NS-2_Trace_Formats

[3] TDMA Documentation
https://www.techtarget.com/searchnetworking/definition/TDMA

[4] CSMA Documentation
https://en.wikipedia.org/wiki/Carrier-sense_multiple_access_with_collision_avoidance

[5] Analysis of Hybrid CSMA/CA-TDMA Channel Access Schemes with Application to Wireless Sensor Networks
https://mspace.lib.umanitoba.ca/bitstream/handle/1993/22284/Shrestha_Bharat.pdf?sequence=1

[6] Differences between CBR and FTP
https://networkengineering.stackexchange.com/questions/24882/difference-between-cbr-and-ftp