University of Waterloo

Faculty of Engineering

Department of Electrical and Computer Engineering

ECE 358

Laboratory 2

Prepared by

Jinming Zhang

UW Student ID Number: 20613667

UW User ID: Jm3zhang @edu.uwaterloo.ca

3B Computer Engineering

and

RuiZhou Xu

UW Student ID Number:

UW User ID: @edu.uwaterloo.ca

3B Computer Engineering

14 March 2019

Table of content

Explanation of our code: ……………………………………………………3

Question 1 …………………………………………………………………..8

Question 2……………………………………………………………………12

# Explanation of our code:

A screenshot of a cell phone

Description automatically generated

In our code, we did not generate the nodes with all the packets at the beginning. We only generate the first packet for each node initially.



Stacked\_packet\_counter is used to keep track of the packets that takes place at the time period of Twait. Nodes\_collision\_counter is used for the collision counter of exponential backoff. Busy\_counter is the exponential backoff counter used for busy states at the non-persistent mode. AllTransmittionDone is used to check whether the time meet the Tsim.



First, we find the nodes that will collide with sender.



Then, we deal with the busy states based on the mode.

A screenshot of a cell phone

Description automatically generated

If there is collision, we use handle\_collision to do the exponential backoff for each packets. If there is no collision, we reset the counter and generate new packets for the sender. For both situations, we need to increment the counter for transmitted and successfully\_transmitted correcpondingly.



After, that we get the efficiency and throughput.

A screenshot of a cell phone

Description automatically generated

In collision detector, we go through each node in the network and find out all of the nodes that are involved in the collision based on the Tsender\_start + Tprop\*distance.

A screenshot of a cell phone

Description automatically generated

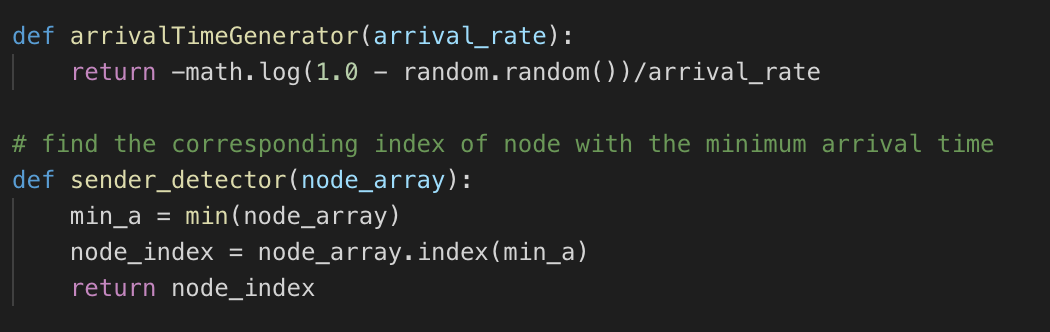
In handle busy persistent, we check whether the arrival time is busy, and use pushed\_time and stacked\_packet\_counter to store the number of packets that take place during the transimission time.

A screenshot of a cell phone

Description automatically generated

In collision handler, we also use pushed\_time and stacked\_packet\_counter to store the number of packets that take place during the propagation time. We also use the exponential backoff to add Twait for each collided packets.

Following is some other helper functions:



In the end, we have verified the our code by comparing the output for efficiency when A = 5, 12 for the persistent CSMA/CD protocol.

Looks similar to the one on the lab manuel.

**In all questions, distance between two neighboring nodes is D = 10 meters and the**

**propagation speed is S = (2/3) \* C 𝑚/𝑠𝑒𝑐, where C = 3 \*10, 𝑚/𝑠𝑒𝑐 is the speed of light.**

# Question 1

**Simulate a persistent CSMA/CD protocol. Show the efficiency and throughput (in**

**Mbps) of the LAN as a function of N (20, 40, 60, 80, and 100) for A = 7, 10 and 20**

**Packet/sec, R = 1 Mbps, and L = 1500 bits. Comment on the behavior of the graphs.**

Simulate a persistent CSMA/CD protocol:

R = 1000000

L = 1500

D = 10

S = 2\*100000000

T\_sim = 1000

for A in [7, 10, 20]:

print("we have a new A:"+str(A)+" here")

for N in [20,40,60,80,100]:

print("we have a new N:"+str(N)+" here ")

persistent(N, A, R, L, D, S, T\_sim)

We are taking A as [7, 10, 20] and N as [20,40,60,80,100] to simulate the CSMA/CD protocol with the CSMA\_simulator function.

Some sample outputs are:

we have a new A:7 here

we have a new N:20 here

149327

140267

efficiency:0.9393277839908389

throughput:0.2104005

trans:0.0015

prop:5e-08

we have a new N:40 here

370725

280227

efficiency:0.7558891361521344

throughput:0.4203405

trans:0.0015

prop:5e-08

With all the inputs, we construct summary tables:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Efficieny |  | | | A |  |
| N | 7 | 10 | 20 |
| 20 | 92.2665414 | 87.14651206 | 62.29852505 |
| 40 | 74.89950423 | 57.52872799 | 28.13725306 |
| 60 | 52.78601659 | 32.85557015 | 19.33201945 |
| 80 | 30.22430404 | 21.14480032 | 17.171943 |
| 100 | 19.20892094 | 17.08194979 | 15.6303408 |

|  |  |  |  |
| --- | --- | --- | --- |
| Throughput | | A |  |
| N | 7 | 10 | 20 |
| 20 | 20.93355 | 29.9508 | 59.96445 |
| 40 | 41.97855 | 59.93685 | 100 |
| 60 | 62.95725 | 87.4719 | 100 |
| 80 | 81.5574 | 100 | 100 |
| 100 | 93.0852 | 100 | 100 |

And We obtain the graphs:

Comment on the behavior of the graphs:

For the Efficiency of the persistent CSMA/CD protocol graph, we can see that as the number of the nodes (N) increased, the efficiency of the persistent CSMA/CD protocol will decrease accordingly. Also the efficiency seems to approach 15% as its final value when number of the nodes increased. This might due to the fact that as the number of the nodes increased, it will increase the likelihood of the amount of collision happens. As the number of the collision increases, the amount of the transmitted packets will increase but left the amount of successfully transmitted packets grow at a slower pace. This will decrease the efficiency will agree with the trend on the graph. On the other hand, as the arrival rate (A) increase, the time stamp between each arrival packet will decrease which means that there will be more packets arrived during the simulation period. This will result a long wait time due to the exponential back off which leads to a decrease in the efficiency. This also agree with the trend on the graph.

For the Throughput of the persistent CSMA/CD protocol graph, we can see that as the number of the nodes increased, the efficiency of the persistent CSMA/CD protocol will increase accordingly. Also the throughput will have capped at 100% as its final value when number of the nodes increased. This might due to the fact that as the number of the nodes increased, the amount of successfully transmitted packets will also increase.

Since the throughput is proportional to the number of the successfully transmitted packets, the throughput will increase as the number of the nodes increase. Similarly, as arrival rate (A) increase, the time stamp between each arrival packet will decrease which means that there will be more packets arrived during the simulation period. This will also increase the amount of successfully transmitted packets and the throughput will increase respectively. This agree with the trend on the graph.

Side notes: From our data, the throughput for high value of N and A will exceed 1. Since it already achieved the best performances

## Stability Test:

For 2T(T = 2000):

As we can see here, this graph looks exactly the show as the previous one. The simulator is pretty stable.

# Question 2

**Show the efficiency and throughput (in Mbps) of non-persistent CSMA/CD protocol for the same network parameters used in question 1. Also, comment on graph and compare between the results obtained in question 1 and question 2.**

Simulate a persistent CSMA/CD protocol:

R = 1000000

L = 1500

D = 10

S = 2\*100000000

T\_sim = 1000

print("here is non-persistent")

for A in [7, 10,20]:

print("we have A:"+str(A))

for N in [20,40,60,80,100]:

print("we have a new N:"+str(N))

CSMA\_simulator(N, A, R, L, D, S, T\_sim, False)

We are taking A as [7, 10, 20] and N as [20,40,60,80,100] to simulate the CSMA/CD protocol with the CSMA\_simulator function.

Some sample outputs are:

here is non-persistent

we have A:7

we have a new N:20

transmitted:140168

successfully transmitted:140151

efficiency:0.9998787169682096

throughput:0.2102265

trans:0.0015

prop:5e-08

we have a new N:40

transmitted:279913

successfully transmitted:279752

efficiency:0.999424821283756

throughput:0.419628

trans:0.0015

prop:5e-08

With all the inputs, we construct summary tables:

|  |  |  |  |
| --- | --- | --- | --- |
| Efficieny | A | | |
| N | 7 | 10 | 20 |
| 20 | 92.2665414 | 87.14651206 | 62.29852505 |
| 40 | 74.89950423 | 57.52872799 | 28.13725306 |
| 60 | 52.78601659 | 32.85557015 | 19.33201945 |
| 80 | 30.22430404 | 21.14480032 | 17.171943 |
| 100 | 19.20892094 | 17.08194979 | 15.6303408 |

|  |  |  |  |
| --- | --- | --- | --- |
| Throughput | | A |  |
| N | 7 | 10 | 20 |
| 20 | 20.93355 | 29.9508 | 59.96445 |
| 40 | 41.97855 | 59.93685 | 100 |
| 60 | 62.95725 | 87.4719 | 100 |
| 80 | 81.5574 | 100 | 100 |
| 100 | 93.0852 | 100 | 100 |

And We obtain the graphs:

Comment on the behavior of the graphs:

For the Efficiency of the non-persistent CSMA/CD protocol graph, we can see that as the number of the nodes (N) increased, efficiency stays relatively high. As N increase, efficiency will decrease. Similarly, as A increase, efficiency will also decrease. The reason of that will be the same as mentioned in the persistent report.

For the Throughput of the non-persistent CSMA/CD protocol graph, we can see that as the number of the nodes increased, the efficiency of the non-persistent CSMA/CD protocol will increase accordingly. Also the throughput will have capped at 100% as its final value when number of the nodes increased. This might due to the fact that as the number of the nodes increased, the amount of successfully transmitted packets will also increase. Since the throughput is proportional to the number of the successfully transmitted packets, the throughput will increase as the number of the nodes increase. Similarly, as arrival rate (A) increase, the time stamp between each arrival packet will decrease which means that there will be more packets arrived during the simulation period. This will also increase the amount of successfully transmitted packets and the throughput will increase respectively. This agree with the trend on the graph.

As we can see from the plots for the non-persistent, when the arrival rate remains the same, the efficiency maintains high when the number of nodes increases. Compared with the persistent plots, the efficiency of the non-persistent is higher when there are more nodes in a network. The reason is that exponential back off is also used for the busy condition in non-persistent mode. In this case, the waiting time of the node will increase when the number of “busy state” increases. With an increasing waiting time, the node will try less attempts to transmit the packets. However, the persistent mode will keep trying to transmit the packet. Therefore, the transmitted packets of non-persistent mode is fewer than the persistent mode and it makes the non-persistent mode has a higher efficiency.

## Stability Test:

For 2T(T = 2000):

As we can see here, this graph looks exactly the show as the previous one. The simulator is pretty stable.