# CECS 474: Computer Network Interoperability

# Project 2: CSMA/CD Performance Evaluation

## Table of Contents

1. Description ..... 3
2. CSMA/CD Overview ..... 3
   2.1 Persistent (1-persistent) CSMA/CD ..... 4
   2.2 Non-Persistent CSMA/CD ..... 4
3. Simulation ..... 5
   3.1 Assumptions ..... 5
   3.2 Network Parameters ..... 5
   3.3 Simulation guide for persistent CSMA ..... 6
   3.4 Results trend ..... 9
4. Questions ..... 10
5. Final Report ..... 10

# 1. Description

The learning objective of this project is to design and implement a ***discrete event simulator*** to evaluate the performance of local area networks (LAN) using CSMA/CD protocols.

# 2. CSMA/CD Overview

An outline of the CSMA/CD protocol, as discussed in class, is shown in Figure 1.

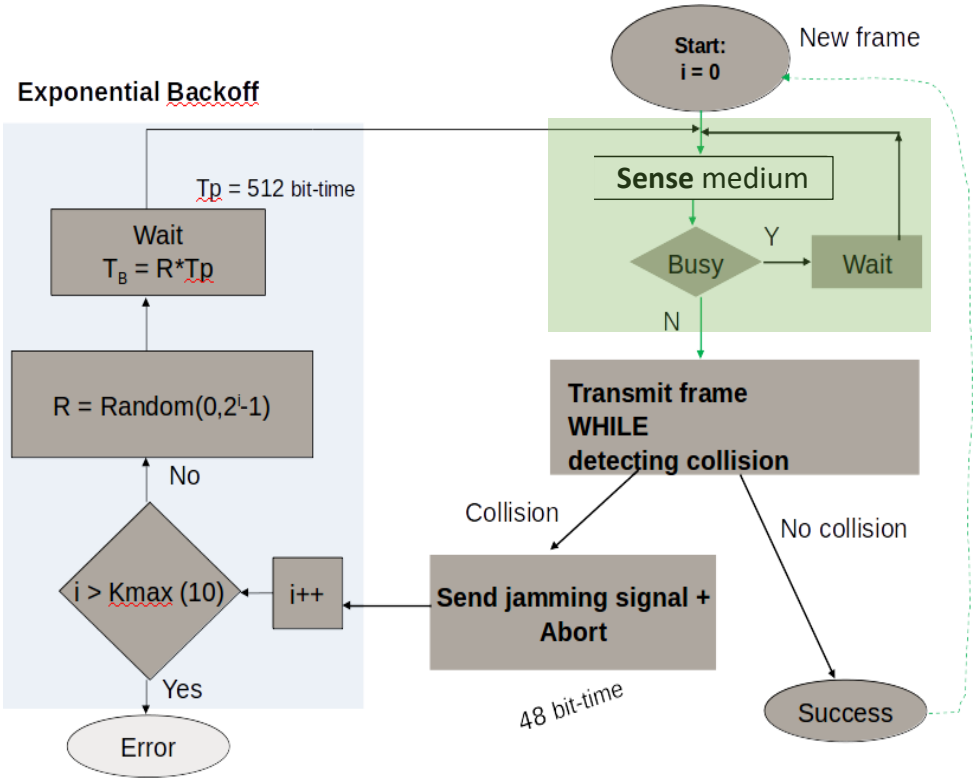


Figure 1: Flow diagram of CSMA/CD (The "Wait" time after "Busy" is equal to the "Wait" time in exponential backoff.)

We will study persistent and non-persistent CSMA/CD protocols. The difference between these protocols is the implementation of the green box at the top right of Figure 1.

## 2.1 Persistent (1-persistent) CSMA/CD

When a node has data to send, it first listens on the channel to see if anyone else is transmitting at that moment. If the channel is idle, the node sends its data. Otherwise, if the channel is busy, the node keeps sensing the channel until it becomes idle. Then, the node transmits a frame. If a collision occurs, the node waits a random amount of time and starts all over again. The protocol is called 1-persistent because the node transmits with a probability of 1 when it finds the channel idle. The green box at the top right of Figure 1 is implemented as shown in Figure 2.

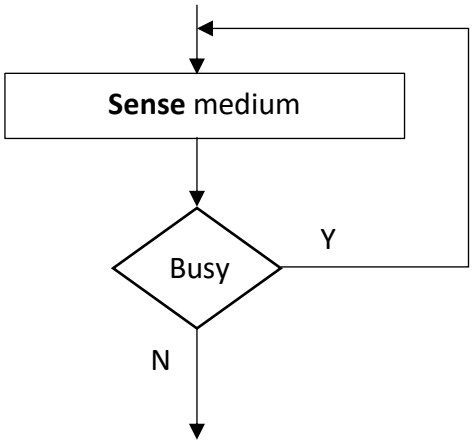


Figure 2: Flow diagram of 1-persistent CSMA/CD

## 2.2 Non-Persistent CSMA/CD

In non-persistent CSMA/CD protocol, a node has less greedy attitude than that in the persistent CSMA/CD. In non-persistent CSMA/CD protocol, a node senses the channel when it has a frame to send, and if the channel is idle, the node begins its transmission. However, if the channel is busy, the node does not continuously sense it to seize it immediately upon detecting the end of the previous transmission. Instead, it waits a random period of time (similar to the exponential backoff) and then repeats the algorithm. The green box at the top right of Figure 1 is implemented as shown in Figure 3.

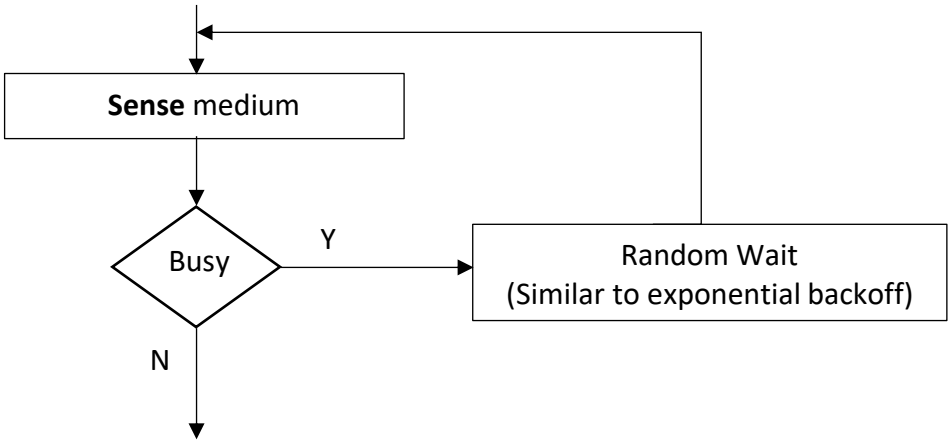


Figure 3: Flow diagram of non-persistent CSMA/CD

# 3. Simulation

## 3.1 Assumptions

1. You are not required to implement the function of the jamming signal in the case of a collision.
2. Once a collision happens, all the nodes will discover it immediately.
3. All the nodes are equally spaced on the bus/channel, i.e., the distance between any two adjacent nodes is the same.
4. The random wait time before sensing the bus/channel in the non-persistent CSMA/CD protocol is the same as the wait time for the exponential backoff.
5. You may use any of the following programming languages: C/C++, Python, Java.

## 3.2 Network Parameters

The network parameters that you will need for simulating CSMA/CD are as follows:

- N: The number of nodes/computers connected to the LAN (variable).
- A: Average packet arrival rate (packets/second) (variable). Data packets arrive at the MAC layer following a Poisson process at all nodes.
- R: The speed of the LAN/channel/bus (fixed).
- L: Packet length (fixed).
- D: Distance between adjacent nodes on the bus/channel.
- S: Propagation speed.

## 3.3 Simulation guide for persistent CSMA/CD

*The following is one possible method to implement the persistent CSMA/CD. You are free to follow this method or any other methods of your choice.*

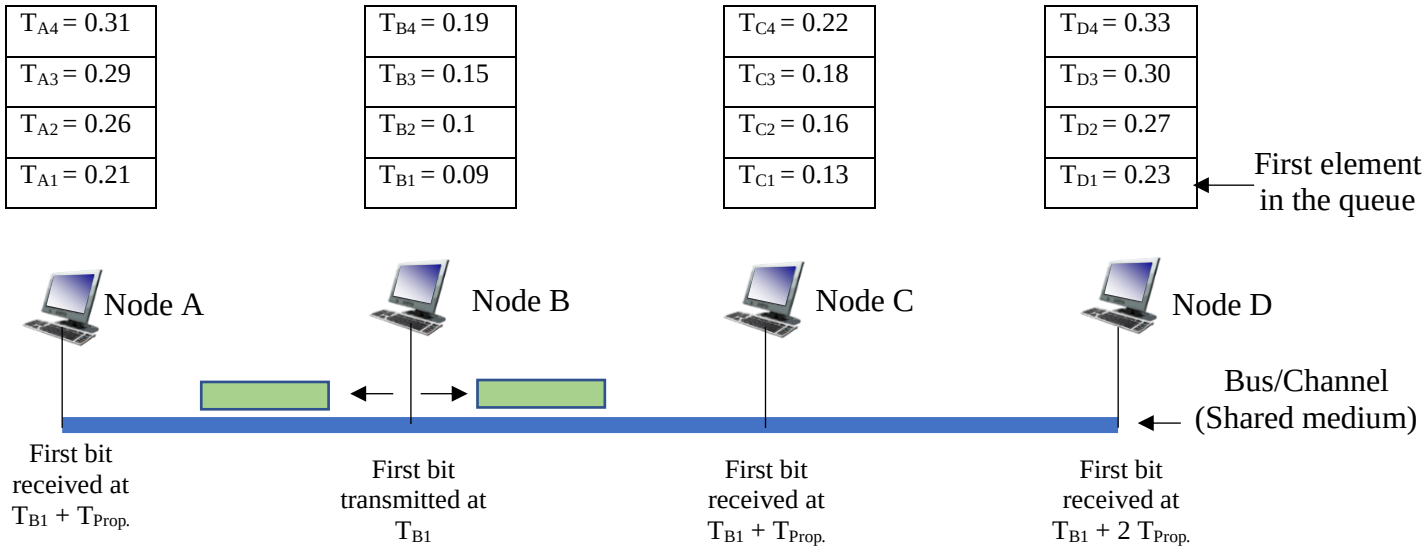


Figure 4: Simulation scenario for persistent CSMA/CD

- All the nodes are equally spaced on the bus/channel, i.e., the distance between any two adjacent nodes is the same.
- Each node on the bus will have its own queue.
- Select a simulation time $T_{sim}$, you can start with $T_{sim}$ = 1000 sec.
- Generate a set of packet arrivals for each node.
- Figure 4 shows a simulation scenario for persistent CSMA/CD with some hypothetical packet arrival times in each node.
- In order to select which node will transmit first, compare the first packet arrival time in all the nodes and select the node with the smallest packet arrival time to be the sender.
- In the simulation scenario shown in Figure 4, Node B will be the sender as it has the smallest arrival time $T_{B1}$ = 0.09 Sec. You will need to maintain a counter of the transmitted packets. This counter will be used to calculate the efficiency of CSMA/CD.
- Node B will transmit the first bit of this packet at $T_{B1}$ = 0.09 Sec. However, due to propagation delay ($T_{Prop.}$), the first bit of this transmitted packet will arrive at Node C at $T_{B1} + T_{Prop.}$, at Node D at $T_{B1} + 2\ T_{Prop.}$ and at Node A at $T_{B1} + T_{Prop.}$. In other words, each node will see the bus busy at different (may be overlapping) durations of time. In the following steps, we will consider that Node B is the sender.
- In order to determine if a collision will happen or not, you need to scan all the nodes except the transmitting node to check if there are any packets to be transmitted before

the arrival of the first bit of the packet (from node B) already travelling on the bus/channel. For example, if the first arrival packet at node C happens at $T_{C1}$ and $T_{C1} <= T_{B1} + T_{Prop.}$, then in this case a collision will happen. If this is the case, increment the number of transmitted packets. Also, you should maintain a collision counter for each node to count the number of collisions it experienced. This collision counter will be used to calculate the exponential backoff time for each node. The collision counter of a node should be reset after a successful transmission by that node.

- As mentioned in the simulation assumptions section, you are not required to implement the function of the jamming signal in the case of a collision.
- If a collision occurred, you need to apply the exponential backoff as shown in Figure 1 to calculate the waiting time $T_{waiting}$ for each node that was part of the collision. During the waiting time of a node, the node is not allowed to transmit any packet. Hence, you need to update the packets arrival times in this node's queue as follows: if those packet arrival times are less than the waiting time, update those arrival times to be equal to the end of the node's waiting time. In other words, if a packet arrives during the waiting time of a node, it should be considered to be arriving at the end of the waiting time.
- If the number of collisions a node experiences while trying to transmit a packet is greater than 10 collisions, this packet should be dropped (i.e., removed from the queue) and the collision counter for this node should be reset.
- If no collision occurred, i.e., a packet is successfully transmitted, the packet has to be removed from the sender's queue. You need to maintain a counter of the number of successfully transmitted packets. This counter will be used to calculate the efficiency of CSMA/CD. Now, you need to scan every node's queue to update the arrival times of the packets as follows. If a packet arrives at a node's queue during the period of time this node sees the bus busy, i.e., a packet is being transmitted on the bus, then those arrival times should be updated to the time of receiving the last bit of the transmitted packet. For example, if the first packet arrival at node C happens at $T_{C1}$ and $T_{B1} + T_{Prop} < T_{C1} < T_{B1} + T_{Prop} + T_{transmission\_delay}$, where, $T_{transmission\_delay}$ is the transmission delay of a packet and $T_{B1} + T_{Prop} + T_{transmission\_delay}$ is the time of receiving the last bit of the packet transmitted by Node B, then, $T_{C1}$ should be set to equal to $T_{B1} + T_{Prop} + T_{transmission\_delay}$. The main reason behind this time update is that if node received packet while the bus/channel is busy, this node will buffer the packet until the bus/channel is free according to CSMA/CD protocol.
- At the end of the simulation, you can calculate the efficiency using the total number of packets and the total number of successfully transmitted packets. You will have to consider the dropped packets due to collisions.
- Repeat your simulation and check whether the variations in your results are within 5% or not. If not, increase the simulation $T_{sim}$.

### 3.4 Simulation guide for non-persistent CSMA/CD

For the non-persistent CSMA/CD simulation, you may follow the same simulation guidelines as those for the persistent CSMA/CD. However, you will need to change the way of updating packets arrival times as follows: If a packet arrives at a node's queue during the period of time this node senses the bus busy, i.e., a packet is being transmitted on the bus, then those arrival times should be updated to be the arrival time of this packet plus a random

waiting time calculated using exponential backoff. For example, in the simulation scenario in Figure 4, if node B started transmitting a packet at time $T_{B1}$, and the first packet arrival at node C happens at $T_{C1}$ and $T_{B1} + T_{Prop} < T_{C1} < T_{B1} + T_{Prop} + T_{transmission\_delay}$, then, $T_{C1}$ will increment its collision counter and update the packet arrival time to be $T_{C1} + T_{backoff\_delay}$, where $T_{backoff\_delay}$ is the exponential backoff. After $T_{C1} + T_{backoff\_delay}$, Node C will sense the bus again and if the bus is busy again, it will increment its collision counter and update the packet arrival time. The collision counter will be reset after the bus is sensed idle. If the collision counter reaches its maximum value of 10 without successfully transmitting the packet, the packet will be dropped.

It should be noted that in non-persistent CSMA/CD, each node will have two collision counters: one for the random wait time before transmission when the bus is busy, and the second collision counter is for the random wait time after a node experiences a collision. At the end of the simulation, you can calculate the efficiency using the total number of packets and the total number of successfully transmitted packets. You will have to consider the dropped packets due to collisions and those dropped due to finding the bus busy.

## 3.5 Results trend

Figure 5 shows the simulation results using the same network parameters as those used in question 1 in the following section for persistent CSMA/CD with packet arrival rates 5 and 12 packets/sec. Using Figure 5, you can judge your simulation results.

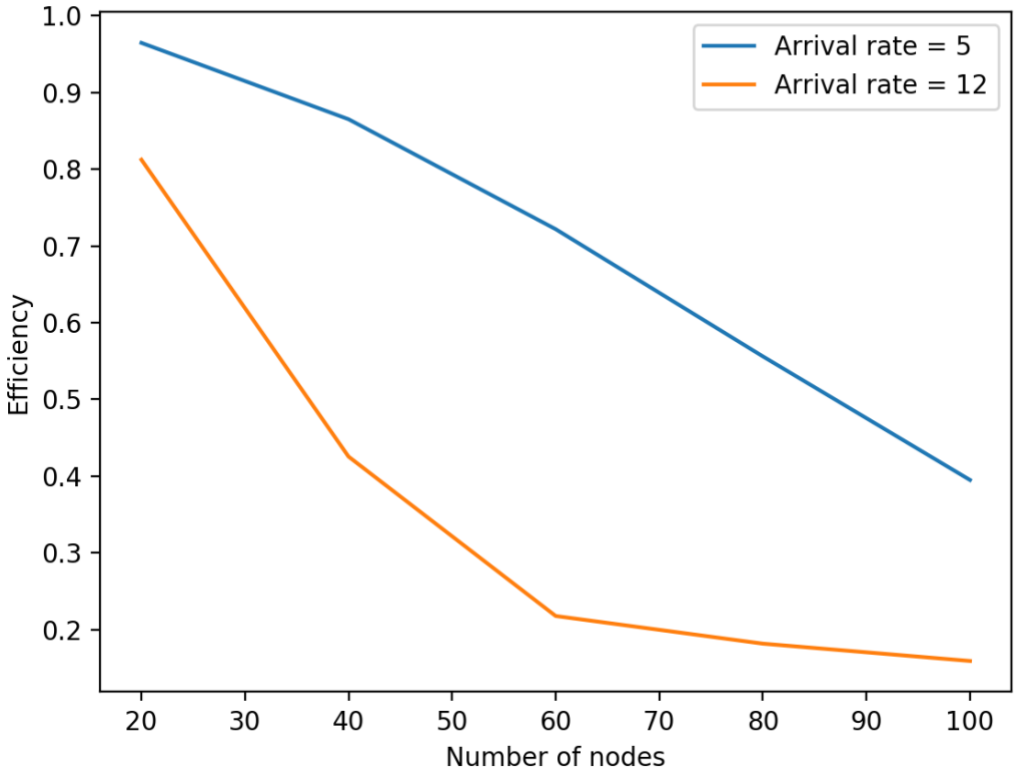


Figure 5: Simulation results for persistent CSMA/CD with packet arrival rates 5 and 12 packets/sec

## 4. Questions

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 × 10^8 $m/sec$ is the speed of light.

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. Show your code in the report. In particular, define your variables. Should there be a need, draw diagrams to show your program structure. Explain how you compute the performance metrics.

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.

## 5. Final Report

Submit the following to the dropbox on BeachBoard:

1. A complete report with
    a. Table of contents
    b. Answer all the questions and provide explanations as asked for.

2. Source code with proper documentation/comments. Insufficient documentation will result in losing marks. Include a makefile that builds and runs your code.

You may be asked to give a demo of your simulator.

Your report, which must include a description of the design of your simulator, all the assumptions, performance graphs, and comments on the graphs.