Exercise 2: Simulating Medium Access Control (MAC) in Octave

Mobile Networking (Communication Networks III)
Secure Mobile Networking Lab - SEEMOO
Matthias Hollick, Allyson Sim, Dingwen Yuan, and Robin Klose
November 09, 2018

Goal and Learning Objectives

- This simulator design exercise tries to familiarize you with both MAC and the use of Octave simulator. It covers the protocol to access the channel and exponential backoff algorithm.
- In particular, you will simulate an event-based basic contention-based MAC access and RTS/CTS-based access as described in Bianchi's paper available in [1]. To be considered in the bonus system.

Schedule

You should complete the following task in order be considered for the bonus system.

a) Design the simulator based on the provided skeleton code for both Task I and Task II and submit the codes (in .m files) for generating the simulator.

b) Submit a digital report (in PDF) consisting of all the tasks (the details are provided under each task.)

c) Each of you must submit an individual report.

d) The codes and report must be submitted before 13:00 on 23 November 2018.

Installing Octave

To begin this exercise, you must first download and install Octave from the following website:
https://www.gnu.org/software/octave/

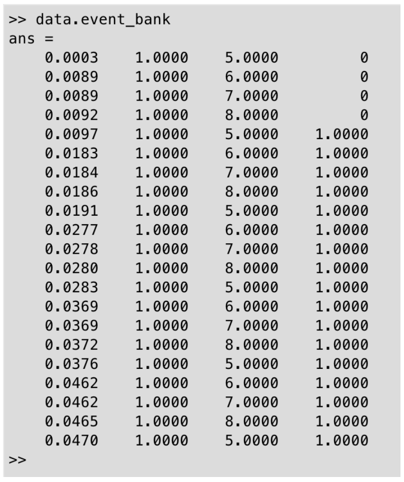
Skeleton code

You are provided with the codes that generates the basic contention-based MAC with one node. This code consists of a function code (function_input_data) that provides all the necessary parameters including the transmission time for different frames and other initialization index.

Once Octave is installed, place the octave files (main_basic and function_input_data) in the same folder and execute/run the file named main_basic. Please enter your name and matriculation number in line 10 and 11, respectively in main_basic. When you run the code, it provides two outputs:

Output 1: All the events, i.e., data.event_bank in Fig. 1

Each row of this parameter (data.event_bank) stores all the information regarding the event, which are: the event time (data.event(id.time)) in column 1, the node's index (data.event(id.node)) in column 2, the node's state (data.event(id.state)) in column 3, and the contention window of the node (data.event(id.cw)) in column 4. Since the nodes in a basic contention-based MAC directly transmit the payload frame. The initial state is state.payload in line 22 of main_basic.



```
>> data.event_bank
ans =
    0.0003    1.0000    5.0000         0
    0.0089    1.0000    6.0000         0
    0.0089    1.0000    7.0000         0
    0.0092    1.0000    8.0000         0
    0.0097    1.0000    5.0000    1.0000
    0.0183    1.0000    6.0000    1.0000
    0.0184    1.0000    7.0000    1.0000
    0.0186    1.0000    8.0000    1.0000
    0.0191    1.0000    5.0000    1.0000
    0.0277    1.0000    6.0000    1.0000
    0.0278    1.0000    7.0000    1.0000
    0.0280    1.0000    8.0000    1.0000
    0.0283    1.0000    5.0000    1.0000
    0.0369    1.0000    6.0000    1.0000
    0.0369    1.0000    7.0000    1.0000
    0.0372    1.0000    8.0000    1.0000
    0.0376    1.0000    5.0000    1.0000
    0.0462    1.0000    6.0000    1.0000
    0.0462    1.0000    7.0000    1.0000
    0.0465    1.0000    8.0000    1.0000
    0.0470    1.0000    5.0000    1.0000
>>
```

Fig. 1: Events of transmission with inter-spacing time (IFS) for wifi network with 1 node.

Output 2: A figure depicting the events

In Fig. 2, the x-axis shows the time of each event and the y-axis shows the state or the event at a particular time. This figure is plotted using the events that are shown in Fig. 1.

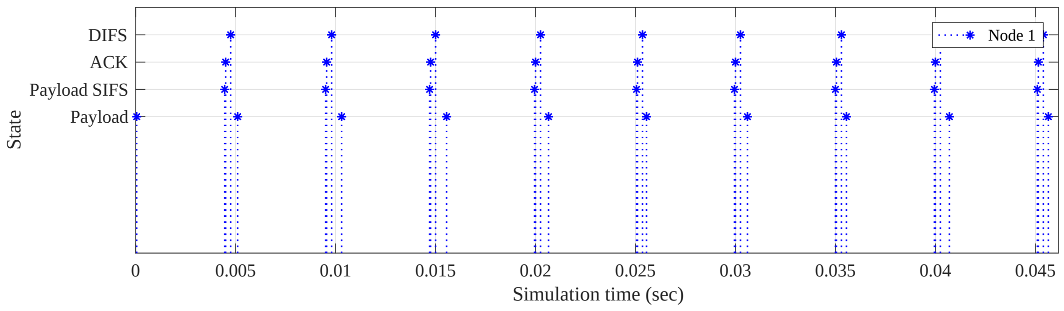


Fig. 2: Stem plot of events of transmission with inter-spacing time (IFS) for wifi network with 1 node.

Task 1 RTS/CTS-based MAC

In this task, extend the provided codes to include RTS and CTS control messages with the following guidelines:

a) add four new states to the simulator. These states are state.rts, state.rts_sifs, state.cts, and state.cts_sifs.

b) since RTS-frame is the first frame, edit the initial state accordingly.

Plot the figure that shows the channel access of the nodes. You can use any plot method to represent the channel access. An example is as shown in Fig. 3. As seen, there are additional states (i.e., RTS, SIFS, CTS, SIFS) added to the plot.

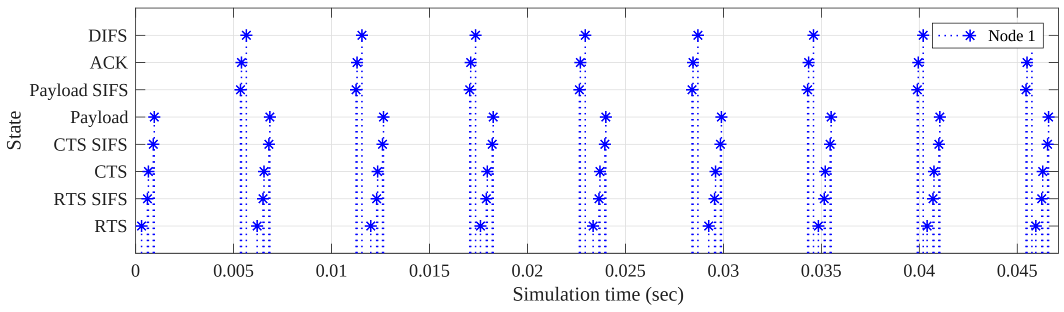


Fig. 3: Stem plot of events of transmission with inter-spacing time (IFS) for wifi network with 1 node for RTS/CTS-based MAC.

To be submitted

a) Simulation code in .m which generates the plot (as in Fig. 3) and the event as in Fig. 1

Task 2 Channel access of multiple nodes

In order to show the contention of multiple nodes within a wifi-network, extend the codes in Task 1 to enable the contention of multiple nodes. Based on Bianchi's paper in [1], and the lecture slides, the following must be included:

a) when a node successfully access the channel before the other nodes, the other nodes must perform backoff accordingly. Specifically, when a node has a new frame to transmit and the medium is sensed busy, an initial backoff counter must be set. Whenever the channel is sensed busy, nodes must stop counting down their backoff counters.

b) if a collision happen, the contention window size must be increased accordingly.

c) upon successful transmission, the contention window size is reset to the minimum, which in this exercise is 8 (in the parameter called input.cw_vec)

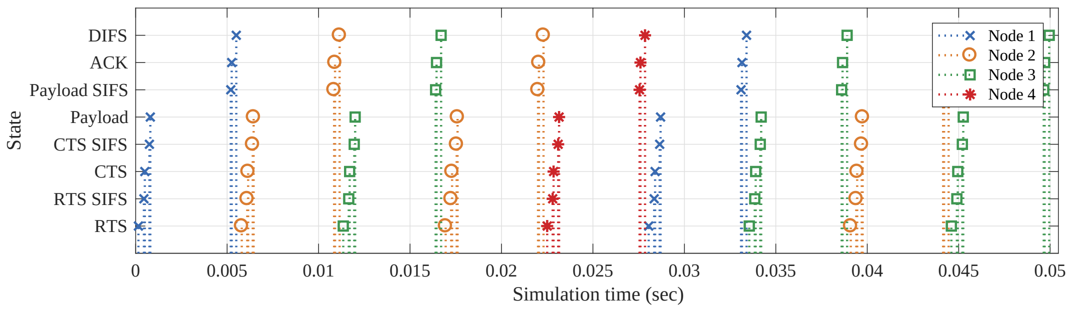


Fig. 4: An example stem plot of events of transmission with inter-spacing time (IFS) for wifi network with 4 node for RTS/CTS-based MAC.

To be submitted

a) Simulation code in .m which generates the plot in Fig. 4 with input.no_nodes = 4 and the event as in Fig. 1.

Task 3 Analysing MAC performance

Set the simulation time to 1s (input.simulation_time = 1). Evaluate impact of the following:

a) Transmit probability (input.tx_prob in line 17 of main_basic) tx_prob = $\{0.01, 0.02, 0.05, 0.1, 0.2\}$, where the default value is 0.2.

b) Frame size (input.packet_payload in line 18 of main_basic packet_payload = $\{1000, 2000, 4000, 8000, 16000\}$, where the default value is 4000 bits.

c) Contention window size (input.cw_vec in line 19 of main_basic by setting the minimum window size as = $\{4, 8, 16, 32, 64\}$, where the default minimum window size is 8.

d) Number of nodes (input.no_nodes in line 16 of main_basic packet_payload = $\{2, 4, 8, 16\}$, where the default value is 4.

on the following:

a) channel efficiency (channel time used to transmit payload / total time)

b) fairness $\frac{\left(\sum_{i=1}^{N} x_i\right)^2}{N * \sum_{i=1}^{N} x_i^2}$ where N is the total number of nodes and x_i is the amount of data bits sent.

Example: To evaluate the transmit probability on the channel efficiency, the performance graph should have the transmit probability on the x-axis and channel efficiency on the y-axis.

To be submitted

a) A report including the discussion on the afore-mentioned parameters and their impacts on channel efficiency and fairness of the a wifi network.

b) The simulation codes in .m files.

Important notes:

- Upon completing this task, you will have eight figures. The 1st figure is transmit probability vs. channel efficiency, the 2nd figure is transmit probability vs. fairness, the 3rd figure is frame size vs. channel efficiency, the 4th figure is frame size vs. fairness, and etc.
- Each parameter should be evaluated 50 times.

References

[1] G. Bianchi, "Performance analysis of the ieee 802.11 distributed coordination function," IEEE Journal on Selected Areas in Communications, vol. 18, no. 3, pp. 535–547, March 2000.