جامعة الملك فهد للبترول والمعادن
King Fahd University of Petroleum & Minerals

KING FAHD UNIVERSITY OF PETROLEUM AND MINERALS

COE540 – Computer Networks

Term Paper

MAC Protocols for Tactical Networks - Survey and Performance Evaluation

Adnan Munir g202110550	Ibrahim Albulushi g202115050
Mahmoud Yassin Mahmoud g202113650	

Submitted to

Dr. Ashraf S. Mahmoud

Table of Contents

1. Introduction

Communication over tactical ad hoc networks is challenging and it's very different from the private networks. Given these networks usually with highly mobile nodes, special consideration must be taken when designing data traffic primarily using broadcast and multicast topologies. The demand for vigorous data communication resulting in low end-to-end delays makes the situation even more complicated. Thus, using standardized well-proven protocols seems to be the best solution for similar cases. On the other hand, these standardized protocols have limited performance that can only be useful during specific situations utilizing assumptions about protocols used on the other layers of the communication protocol.

Carrier Sense Multiple Access (CSMA) is a widely deployed protocol. A type of CSMA known as CSMA with collision avoidance (CSMA/CA) is widely utilized and it is a component of the IEEE 802.11 wireless local area network standard. There has been a lot of research that shows CSMA to be effective to operate well for unicast traffic by employing strategies such as automatic repeat requests (ARQ) and channel reservations. These solutions, on the other hand, are ineffective for multicast traffic, because they require the destination to continue providing acknowledgments, this becomes impractical when there are several receivers [1].

In several research on broadcast in ad hoc networks, flooding with an effective choice of relay nodes is employed to tackle multicast routing, the multipoint relay (MPR) approach, as defined by the Simplified Multicast Forwarding (SMF) architecture, is one of the most prevalent ways of identifying relays. Simplified Multicast Forwarding integrates the optimal flooding principle to the data forwarding plane, resulting in a multicast forwarding capability that is suited for use scenarios where localized efficient flooding is a proper design strategy [2]. The network simulations in most of the experiments usually end up making use of somewhat simple channel models, under such assumptions issues such as carrier sensing distances and throughput become far too simple to reflect genuine conditions, particularly in challenging terrains.

In this paper, we compare three different backoff techniques which are the 802.11 protocol, Additive Increase Multiplicative Decrease (AIMD), and Multiplicative Increase Multiplicative Decrease (MIMD). the comparison in the simulation section delivers three main parameters which are packet loss, utilization, and frame delay for

each of the three techniques. Furthermore, the paper illustrates the background in section two, after that in section 3, the article shows the related work in the past recent years. Section four explains the results and outcomes of the used methodology,and at last section ,five concludes the work done.

2. Background

It is tempting to depend on a theoretical CSMA analysis of a 1-persistent network that most network engineers learned early in their careers when judging the value of deploying CSMA-style networks. However, it is vital to comprehend the assumptions made in this theoretical analysis and determine whether they are true in the tactical situation.

The following fundamental assumptions should be considered [3]:

1- A collision on a CSMA network results in no nodes receiving the signal correctly.

2- All nodes are fully aware of when the network is congested.

3- Each broadcast has just one chance to be received.

4- When all units having data to transfer detect that the network is free, they will transmit.

Firstly, we going to discuss the theoretical CSMA case, and then we are going to discuss the case of using EPLRS CSMA. In the next section, we are going to introduce the (CSMA) for transmitting broadcast traffic in mobile military ad hoc networks using multipoint relay (MPR) flooding on the network layer.

2.1. The Theoretical CSMA Case:

CSMA is a MAC protocol that operates as follows:

when a unit has data to transmit, it listens to check whether the network is busy.

if the network is not busy, data can be sent.

if the network is busy, it waits until the network becomes free before sending data.

In a 1-persistent CSMA channel with minimal propagation delay, the theoretical channel throughput S as a function of channel traffic G is given by:

$$S = \frac{G \cdot [1 + G] \cdot \mathrm{e}^{-G}}{G + \mathrm{e}^{-G}}$$

Using this equation, a throughput versus channel traffic curve looks like this:

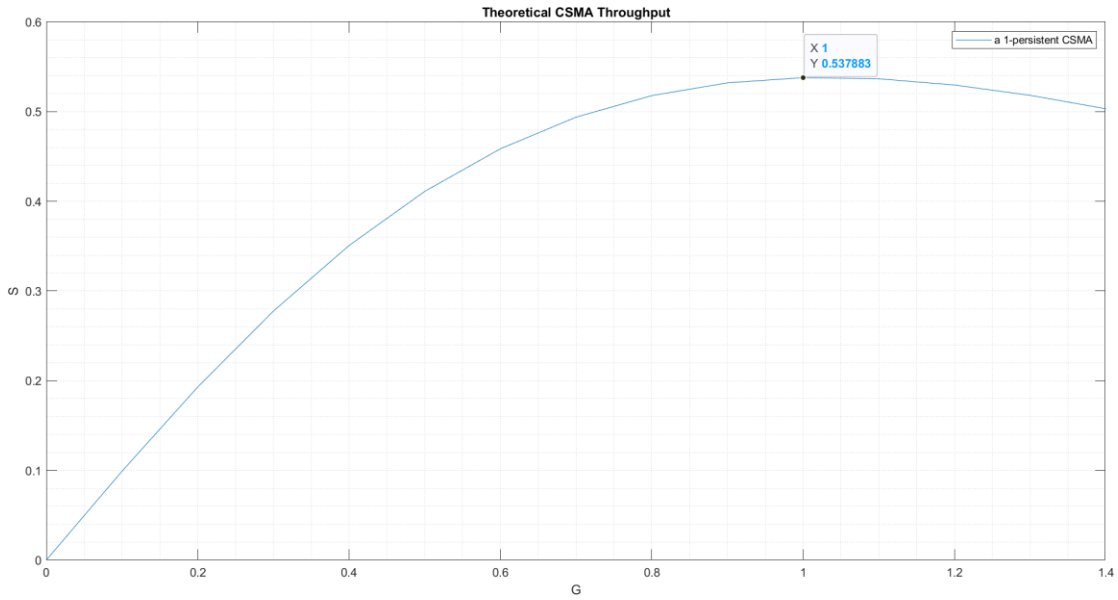

Figure 1. Theoretical CSMA Throughput Curve

In the case of modest loads, throughput increases as load increases.

At a load of one, the theoretical maximum possible throughput in a 1-persistent CSMA network is about 0.54.

It's also worth noting that once the load rises over this amount, the throughput begins to fall.

2.2. The EPLRS CSMA Case:

The EPLRS radio is CSMA capable. It, too, is 1-persistent from the sender and will broadcast only when the network is not busy, like in the theoretical case.

There are, however, major variations in comparison to the assumptions mentioned at the beginning of this section, As follows [3]:

THEORETICAL CSMA	EPLRS
Collision results in loss of all colliding packets	Collision may or may not result in loss - depends on Simal-to-Noise Ratio
All nodes have perfect knowledge of the network's busy	Network busy determined by receipt of valid Transmission Unit from another node
Only one opportunity to receive packets	Even though the source only transmits data once, a series of relays provide nodes extra receive opportunities
All nodes with data to send will send data as soon as the network is not busy	Random wait time after detecting the network is not busy is based on network utilization and Priority

Table 1. the different assumptions between Theoretical and EPLRS CSMA Protocol

2.3. An Important Mathematical Example:

Firstly, for the case of theoretical CSMA, throughput is the same at all nodes on the CSMA network. In other words, each broadcast is received or not received by all nodes. A network collision is the only thing that can prevent communication from being received. As a result, if 10 messages are delivered and none of them clash, throughput is judged to be 1.0. If eight of the ten do not collide and two do, then the throughput is 0.8 times the supplied load. If the network is 1000 bps and the offered load is 100 bps, the throughput is 0.08 or 80 bps. The message completion rate would be 80% and would be consistent across all nodes.

Now, for the case of EPLRS, let's assume the following case:
10 nodes
10 messages are sent, 1 from each node
8 of the messages are being managed to be sent without collision from the source and are received by all 9 of the possible receivers
the other 2 messages are sent simultaneously - 5 of the units receive one of these messages, 3 receive the other, and 2 receive none.
Considering a 1000 bps network and a 100-bps average provided load, the throughput for the five nodes that get eight non-collided and one collided message and the three nodes that receive the other would be:

(0.5+0.1+0.3) *100= 90bps (0.09 normalized)

Moreover, their message completion rate would be 90%. The throughput is 80 bps (0.08 normalized) and the message completion rate is 80% for the two nodes that did not receive either of the colliding messages.
If we average this across all nodes, we have a system throughput of:

$$0.09 \times \frac{8}{10} + 0.8 \times \frac{2}{10} = 0.088$$

and a message completion rate of 88 percent on average. This basic example shows how removing assumptions might result in substantially different outcomes.

2.4. Main Types of Data in Tactical Networks and Their Reliability Issue:

Consider the following two categories of data: Command and Control (C2) and Situational Awareness (SA). Command and control data is made up of directed communications such as instructions, alarms, overlays, and so on. Situational Awareness is comprised of regularly updated friendly position reports, hostile location reports, and other position data.

C2 data, in general, need great dependability. When a commander issues an order, he needs to know that it will be received. Thus, CSMA networks with no upper layer dependability built in are likely insufficient.

SA data, on the other hand, needs less dependability. If a node misses a position update at time t, it is insignificant because another update will be provided shortly after. Only if the node misses several consecutive reports or never gets any reports from a certain node would this be a concern.

Instead, then using traditional network measurements like throughput and completion rate to assess the performance of SA, a more relevant statistic called SA Accuracy has been proposed [4].

If we observe the results that have been shown in the work of Kelsch, Geoffrey R. [3] in figure 2, if a network designer decides that the needed completion rate of SA data is 0.8 to obtain the specified SA picture accuracy, then using the theoretical scenario to analyze network performance indicates that the CSMA solution will function up to loads of roughly 0.5. If the predicted real load is greater (say, 0.6), the designer will need to look for a better solution. However, if the theoretical case's assumptions are not accepted and simulation is used to forecast performance, we can demonstrate that loads of up to 0.8 can be handled adequately.

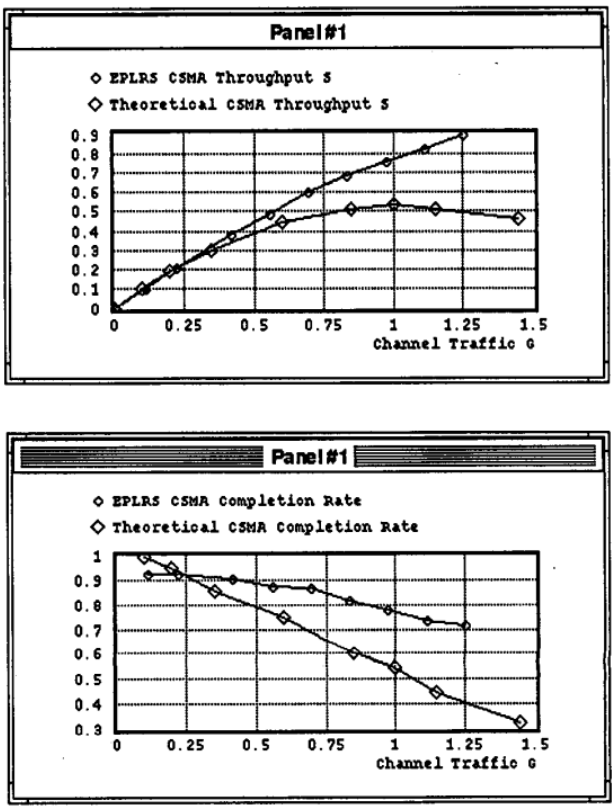

Figure 2. Sample Simulation Results Comparing Theoretical and EPLRS CSMA Cases from the work of Kelsch, Geoffrey R. [3]

The CSMA approach is now sufficient for our estimated load of 0.6 and might be a good design decision.

The network designer should also consider that inside the tactical CSMA network, nodes that are closer together are more likely to receive each other's data a larger proportion of the time (all links are not equal, as the theoretical scenario suggests). This implies that when traffic grows and the network fails, nodes will have a sharper view of what is nearby while the faraway picture will deteriorate. This implies that the soldier may still rely on the short-range image, which might be termed a gentle deterioration.

3. Related Works:

3.1. Additive-Increase Multiplicative-Decrease (AIMD):

The collision channel model is the major channel model used in wireless systems nowadays for Medium Access Control (MAC) protocol designs, IEEE 802.11 [5] is a well-known example for it.

A packet in a collision channel is called successfully received if no other transmission happens in the same slot. This theory, however, does not apply to advanced signal processing techniques like as Code-Division Multiple-Access (CDMA), multiple-antenna arrays, and space-time coding, which allow for the reception of more than one packet at the same time.

The capacity to decode many packets at the same time is known as multipacket reception (MPR). The K-MPR model to describe a wireless channel in which the advanced receiver can decode up to K signals at the same time. If the quantity of current transmissions exceeds K, an interference outage happens, and all packets are destroyed. The commonly understood additive-increase multiplicative-decrease (AIMD) method is a feedback control system being used congestion avoidance in the transport layer (TCP). The pioneering work of Chui and Jain [6] demonstrates that the AIMD algorithm may converge to an efficient and fair state regardless of the starting window size (or access rate).

AIMD has received a great deal of scientific interest and has recently been explored for MAC. Heusse et al. [7] used the AIMD technique to alter the size of the contention window by counting the number of consecutive idle slots between two transmission attempts (CW). Hu et al. [8] presented MAC contention control using a similar strategy (MCC). Rather than altering the size of CW, MCC uses the AIMD technique to update the packet dequeuing rate by monitoring the number of successive collisions or idle slots. These approaches require that all users be within sensing range of one another in order to determine if the channel is idle; else, RTS/CTS-like control messages are required to manage the users.

The appealing work that we have found was done by Li, Ke, et al. [9] which an AIMD-MAC protocol for multiple packet reception channels was suggested. Based on the transmission history, it manages the access probability of each node locally and

regularly. According to their simulation results, it outperforms S-Aloha* under modest traffic loads and reaches the best performance once the system is saturated. They claim that their AIMD-MAC protocol may be effectively used in a dispersed and dynamic wireless environment with the right parameter configuration.

3.2. Some Notes -In General-About:

Multiplicative Increase Multiplicative Decrease (MIMD):

The Fairness in Multiplicative Increase Multiplicative Decrease (MIMD) congestion control algorithm in terms of scalable TCP protocol for high-speed networks was discussed in [10]. A fairness among sessions that share a common bottleneck link and apply the MIMD methodology is investigated. It was found that losses, often known as congestion signals, occur when capacity is exceeded, although they can also occur before that. Both synchronous and asynchronous losses are considered. Only one session suffers a loss at a loss instant in the asynchronous scenario. The article illustrates two main models to identify which source is responsible for a packet loss, the first model is the rate dependent model in which a session's packet loss rate is proportional to its rate at the time of congestion, and the independent loss rate model. The study shows that the capacity is fairly divided in the existence of rate dependent losses, but rate independent losses cause extreme unfairness. Furthermore, the article explained how capacity is divided among sessions with synchronous losses, some of which utilize additive increase multiplicative decrease (AIMD) protocols and others use MIMD protocols. In conclusion, to establish fairness, the arrival rate of these rate dependent tosses must be larger than a specific minimum rate. As a result, in networks implementing MIMD algorithms, a stream of rate-dependent losses would be required to assure equitable sharing. For example, using some buffer management system.

The Distributed Coordination Function (DCF) is based on carrier sense multiple access with collision avoidance (CSMA/CA) and is used extensively in the IEEE 802.11 standard. When a collision occurs in DCF, a backoff mechanism is used to randomly distribute medium access for the nodes. The binary exponential backoff (BEB) technique is the default backoff technique in 802.11. In this study [11] New Backoff Algorithm for IEEE 802.11 MAC Protocol is investigated. The behavior of three different backoff algorithms used in the IEEE802.11 standard are investigated: BEB, I-BEB, and E-BEB, and then the article proposed the New Binary Exponential Backoff (N-BEB) method to increase channel access fairness while maintaining channel

performance and maximum throughput, based on the number of successful and failed transmissions, the N-BEB algorithm enhances the contention window CW selection process. Over BEB, I-BEB, and E-BEB protocols, simulation results show that the proposed protocol improves fairness, reduces missed packets, and has a high transmission rate. The article stated that It's worth noting that the suggested method achieves a significant reduction in end-to-end delay, which was determined to be the shortest among comparing algorithms.

4. Results and Discussion

In this section, we describe the simulation results and parameters used in the simulation. We simulate Carrier Sense Multiple Access with collision avoidance (CSMA/CA) with three different backoff techniques. The exponential backoff, Additive Increase Multiplicative Decrease (AIMD), and Multiplicative increase Multiplicative decrease (MIMD) are three linear congestion control protocols that are applied based on their congestion window (CW). This simulation result is considered hindered number of nodes, packet size 2K, different simulation times in seconds, and 6Mb/s data rate. Table 2 shows some system parameters used in the simulation.

Parameters	**Values**
N (nodes)	100
R (data rate)	6Mb/s
CW	15
Slot time	9μsec
Simulation count	5000

Table 2. Simulation parameters

For the 100 nodes simulation run 1 to 5000 times and we calculate three performance metrics such as packet loss, utilization, and frame delay. When the collision occurs, the packet gets lost, and we set the collision flag to 1. So, for the above-discussed backoff techniques, we deduce the packet loss shown in figure 3.

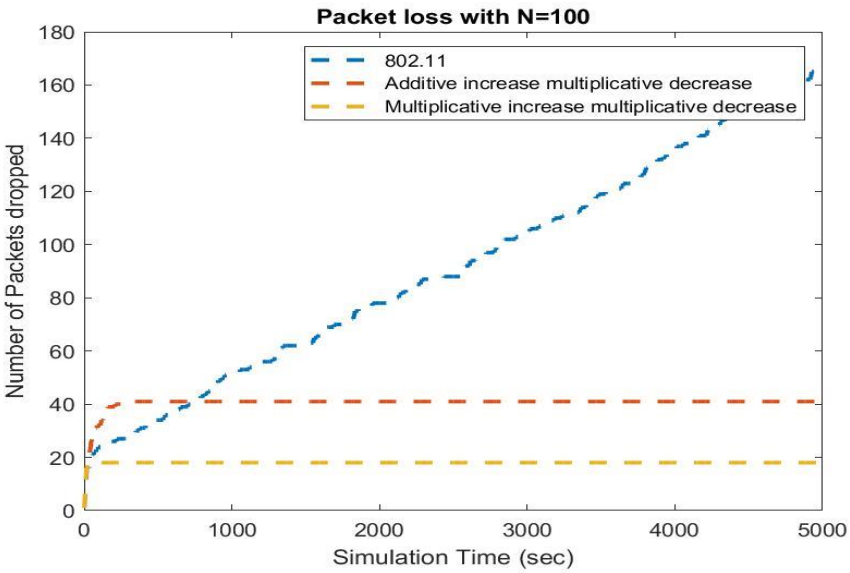

Figure 3. Packet loss

From figure 3 we can see that for the adaptive increase and multiplicative decrease (AIMD) packet loss rate is increasing and after 40 packets it is almost constant. An exponential backoff (802.11) packets are going to drop exponentially. The packet loss ratio goes decreasing for the multiplicative increase and multiplicative decrease (MIMD) because the number of packet losses is going to decrease with time. So MIMD performs better than AIMD and AIMD perform better than 802.11 with respect to packet loss ratio.

A narrower contention window (CW) creates too many collisions for given network size, whereas a bigger window causes fewer transmission attempts. As a result, there is an optimal window size for each network size. The utilization of different CSMA/CA techniques is shown in figure 4. The best utilization for a maximum number of nodes is 99.9% for the MIMD system for 200 nodes. The additive increase and multiplicative decrease have the worst utilization for the system. IEEE802.11 achieves constant utilization for all number of nodes.

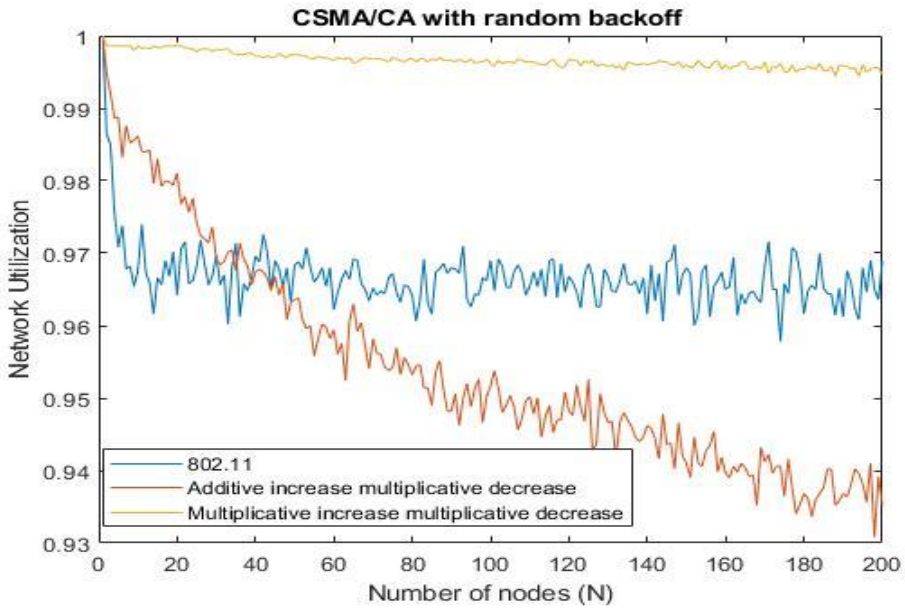

Figure 4. CSMA/CA with random backoff

The third metric is frame delay which we computed. How much time a system takes to send a complete frame without collision. It includes the good time of the system. When a collision occurs, the system will back off according to the above three backoff strategies. It doesn't include collision time. Figure 5 shows the frame delay for all three techniques.

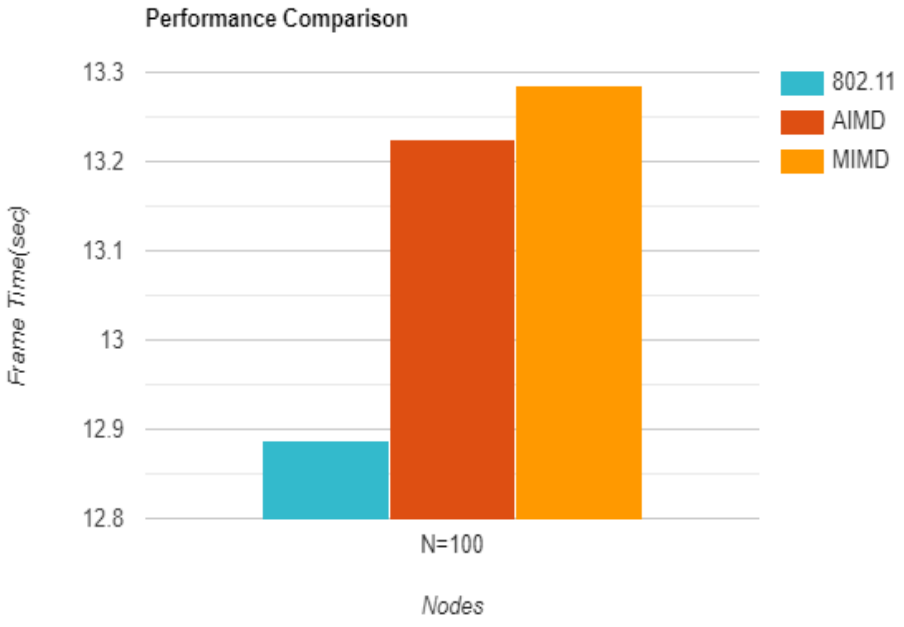

Figure 5. Frame delay

AIMD has a lower frame delay with reference to multiplicative increase and multiplicative decrease (MIMD) has the largest delay time for transferring a complete frame.

From overall results for CSMA/CA Additive increase and adaptive decrease (AIAD) backoff technique performs better for a higher number of nodes and tactical networks in terms of the above three discussed metrics.

5. Conclusion

The model's formulations were validated using packet-level simulations. In this work, we thoroughly explained and simulate the above-discussed backoff techniques for CSMA/CA. MAC protocol. We calculate different performance metrics and from that we can recommend the designer the following:

- If the designer cares more about the utilization and packet loss we recommend him to use the MIMD as the backoff algorithm for his system.
- If the designer cares more about the delay time we recommend him to use the AIMD as the backoff algorithm for his system.

Unfortunately, this is still not good enough for a special broadcasting scenario for tactical networks. For better results for tactical networks, CSMA/CA slot scheduling with real-time reinforcement learning could be used.

6. References

1. Komulainen, J. Grönkvist and U. Sterner, "On the performance of using CSMA for broadcast traffic in tactical ad hoc networks," 2018 International Conference on Military Communications and Information Systems (ICMCIS), 2018, pp. 1-7, doi: 10.1109/ICMCIS.2018.8398718.
2. J. Macker, “Simplified multicast forwarding (SMF),” IETF, Network Working Group, Internet-Draft, January 2012.
3. Kelsch, Geoffrey R. "A comparison of battlefield carrier sense multiple access (CSMA) networks with theoretical CSMA network analysis." MILCOM 1999. IEEE Military Communications. Conference Proceedings (Cat. No. 99CH36341). Vol. 2. IEEE, 1999
4. L.Young, D.A.White, and G.R. Kelsch, “Modeling Situational Awareness in the Tactical Internet,” MILCOM 98 Proceedings
5. LAN/MAN standards Committee, et al.: Information Technology – Telecommunications and Information Exchange Between Systems – Local and Metropolitan Area Networks – Specific Requirements: Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE Std. IEEE (1997)
6. Chiu, D., Jain, R.: Analysis of the increase and decrease algorithms for congestion avoidance in computer networks. Comput. Netw. ISDN Syst. (June 1989).
7. Heusse, M., Rousseau, F., Guillier, R., Duda, A.: Idle sense: an optimal access method for high throughput and fairness in rate diverse wireless lans. In: ACM SIGCOMM (2005)
8. Hu, C., Hou, J.C.: A novel approach to contention control in ieee 802.11e-operated wlans. In: IEEE INFOCOM, pp. 1190–1198 (May 2007)
9. Li, Ke, et al. "Additive-increase multiplicative-decrease mac protocol with multi-packet reception." *International Conference on Wired/Wireless Internet Communication*. Springer, Berlin, Heidelberg, 2013.
10. E. Altman, K. E. Avrachenkov and B. J. Prabhu, "Fairness in MIMD congestion control algorithms," *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, 2005, pp. 1350-1361 vol. 2, doi: 10.1109/INFCOM.2005.1498360.
11. M. Shurman, B. Al-Shua'b, M. Alsaedeen, M. F. Al-Mistarihi and K. A. Darabkh, "N-BEB: New backoff algorithm for IEEE 802.11 MAC protocol," *2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2014, pp. 540-544, doi: 10.1109/MIPRO.2014.6859627.

Appendix

MATLAB Code

```
goodtime = [];
k=1;
fileID = fopen('input_network.txt','r');
formatSpec = '%f';
Input = fscanf(fileID,formatSpec); % Input about relevant info
read %
fclose(fileID);
P_L = 0;
% N = [1:200];%Input(1); % get number of nodes *
N = Input(1);
Packet_size = Input(2) * 8; % get the packet size in bits %
T = Input(3) * 10^-3; % get simulation time in s %
Backoff_St = input('enter strategy')%%Input(4); % get the
random backoff strategy %

if Backoff_St == 1
% Strategy 1 %
 for node = 1:length(N)
    total_time = 0;
    count = 0;
    good_time = 0;
    data_rate = 6 * 10^6; % 6 Mbps %
    packet_time = Packet_size / data_rate; % get packet time %
    slot_size = 9 * (10^-6); % 9 us %
    j = 1;
    simulation_count = 0;
    CW_min = 15;
    CW = 15;
    collision_flag = 0;
    r = (randi([0,CW_min],N(node),1)) * 10^-6; % generate N
random numbers and put them into an array %
    simulation_time = [1:5000];

while total_time < T
     for p=1:length(simulation_time)
        [M,I] = min(r); % find the node with the minimum
counter and index of node %
        simulation_count = simulation_count + 1;

        for i = 1:N(node)   % check if there are more than one
nodes with same minimum counter %
            if (M == r(i))
                count = count + 1;
                collision_index(j) = i;
                j = j + 1;
```

```
            end
        end

        if count > 1
            collision_flag = 1; % collision occured %
            P_L = P_L + 1;
        end

        if collision_flag ~= 1 % if no collision, increase
good time %
            good_time = good_time + packet_time;
            CW = CW_min;
            r(I) = (randi([0,CW],1,1)) * 10^-6;
        else
            CW = CW * 2;
            for i = 1:N(node)
                if (M == r(i)) % for all nodes that collided,
choose new rand %
                    r(i) = (randi([0,CW],1,1)) * 10^-6;
                end
            end
        end

        total_time = total_time + packet_time;

        for i = 1:N(node)
            for j = 1:size(collision_index)
                if(i ~= collision_index(j))
                    r(i) = r(i) - slot_size;
                end
            end
        end
        count = 0;
        collision_flag = 0;
        Loss(p) = P_L;
    end
end
    utility(node) = good_time / total_time;
    goodtime(k) = good_time;
    k=k+1;
 end
figure1 = figure;
axes1 = axes('Parent',figure1)
 plot(simulation_time, Loss,'--','LineWidth',2)
%     figure
%     plot(N,utility)
    hold on
end
```

```
Backoff_St = input('enter strategy')
if Backoff_St == 2
  % Strategy 2 %
  for node = 1:length(N)
  P_L1 = 0;
    total_time = 0;
    count = 0;
    good_time = 0;
    data_rate = 6 * 10^6; % 6 Mbps %
    packet_time = Packet_size / data_rate; % get packet time %
    slot_size = 9 * (10^-6); % 9 us %
    j = 1;
    simulation_count = 0;
    CW_min = 15;
    CW = 15;
    collision_flag = 0;
    r = (randi([0,CW_min],N(node),1)) * 10^-6; % generate N
random numbers and put them into an array %
   while total_time < T
    for p = 1:length(simulation_time)%while total_time < T

        [M,I] = min(r); % find the node with the minimum
counter and index of node %
        simulation_count = simulation_count + 1;

        for i = 1:N(node)   % check if there are more than one
nodes with same minimum counter %
            if (M == r(i))
                count = count + 1;
                collision_index(j) = i;
                j = j + 1;
            end
        end

        if count > 1
            collision_flag = 1; % collision occured %
            P_L1 = P_L1 + 1;
        end

        if collision_flag ~= 1
            good_time = good_time + packet_time;
            CW = round(CW,2);
            r(I) = (randi([0,CW],1,1)) * 10^-6;
        else
            CW = CW + 2;
            for i = 1:N(node)
                if (M == r(i))
                    r(i) = (randi([0,CW],1,1)) * 10^-6;
                end
```

```
            end
        end

        total_time = total_time + packet_time;
        for i = 1:N(node)
            for j = 1:size(collision_index)
                if(i ~= collision_index(j))
                    r(i) = r(i) - slot_size;
                end
            end
        end
        count = 0;
        collision_flag = 0;
        Loss1(p) = P_L1;
    end
   end
    utility1(node) = good_time / total_time;
      goodtime(k) = good_time;
    k=k+1;
  end
    plot(simulation_time, Loss1,'--','LineWidth',2)
% plot(N,utility1)

end

Backoff_St = input('enter strategy')

if Backoff_St == 3
    for node = 1:length(N)
   % Strategy 3 %
    total_time = 0;
    count = 0;
    good_time = 0;
    data_rate = 6 * 10^6; % 6 Mbps %
    packet_time = Packet_size / data_rate; % get packet time %
    slot_size = 9 * (10^-6); % 9 us %
    j = 1;
     P_L2 = 0;
    simulation_count = 0;
    CW_min = 15;
    CW = 15;
    collision_flag = 0;
    r = (randi([0,CW_min],N(node),1)) * 10^-6; % generate N
random numbers and put them into an array %

    while total_time < T
        for p=1:length(simulation_time)
        [M,I] = min(r); % find the node with the minimum
counter and index of node %
```

```
        simulation_count = simulation_count + 1;

        for i = 1:N(node)   % check if there are more than one
nodes with same minimum counter %
            if (M == r(i))
                count = count + 1;
                collision_index(j) = i;
                j = j + 1;
            end
        end

        if count > 1
            collision_flag = 1; % collision occured %
             P_L2 = P_L2 + 1;
        end

        if collision_flag ~= 1
            good_time = good_time + packet_time;
            CW = round(CW,2);
            r(I) = (randi([0,CW],1,1)) * 10^-6;
        else
            CW = CW * 2;
            for i = 1:N(node)
                if (M == r(i))
                    r(i) = (randi([0,CW],1,1)) * 10^-6;
                end
            end
        end

        total_time = total_time + packet_time;
        for i = 1:N(node)
            for j = 1:size(collision_index)
                if(i ~= collision_index(j))
                    r(i) = r(i) - slot_size;
                end
            end
        end
        count = 0;
        collision_flag = 0;
        Loss2(p) = P_L2;
    end
    end
    utility2(node) = good_time / total_time;
      goodtime(k) = good_time;
    k=k+1;
    end
% plot(N, utility2)
    plot(simulation_time, Loss2,'--','LineWidth',2)
end
```

```
legend('802.11', 'Additive increase multiplicative
decrease','Multiplicative increase multiplicative
decrease','Additive increase additive decrease')
xlabel('Simulation Time (sec)')
ylabel('Number of Packets dropped')
title('Packet loss with N=100')
saveas(figure1, 'Packetloss.jpg')
% legend('802.11', 'Additive increase multiplicative
decrease','Multiplicative increase multiplicative
decrease','Additive increase additive decrease')
% xlabel('Number of nodes (N)')
% ylabel('Network Utilization')
% title('CSMA/CA with random backoff ')
% fprintf('Number of Nodes: %d ; Packet Size: %d ; Simulation
Time(s): %d ; Backoff Strategy: %d \n Utilization: ', N,
Input(2), T, Backoff_St);
% saveas(figure1, 'Utilization.jpg')
disp(utility);
color= ['r','g','b','k'];
figure, hold on
% % if data is more than colors then colors will be repeated
m = length(color);
for k = 1:length(goodtime)
    i = mod(k-1,m); %%i is remainder after division of k-1 by
m
    i = i+1;
    h=bar(k,goodtime(k));
    set(h,'FaceColor','flat');
end
legend('802.11', 'Additive increase multiplicative
decrease','Multiplicative increase multiplicative
decrease','Additive increase additive decrease')
ylabel('Frame time (sec)')
```