International Journal of Electronics Communication and Computer Engineering
Volume 4, Issue 2, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

# Evaluation of Simple MAC in NS-2

**Vijaya Sagvekar**
Student Member, NMIMS University, India.
Email: vijayasagvekar@gmail.com

**Vidya Sagvekar**
Electronics Engineering, KJSIEIT, Mumbai.
Email: vidyarsagvekar@gmail.com

*Abstract* – **In this paper performance of Simple MAC for various random wait time is evaluated. Design of efficient medium access control (MAC) protocol with high throughput performance and high degree of fairness performance is major focus in distributed contention-based MAC protocol research. Different MAC mechanisms have been proposed for the decentralized MAC. In this paper, we study 802.11 MAC's Binary Exponential Back off and Simple MAC which uses random wait time for stations. In this paper we demonstrate the behavior of Simple MAC for various random wait times and different scenarios of cross traffic with 4 nodes and simple 2 node scenarios in Network Simulator (NS2).**

*Keywords* **– Contention Window, Jitter, Medium Access Control Protocol, Network Simulator 2, Wireless Communication.**

## I. Introduction

### *1.1 Wireless Communication*

Wireless communication is the transfer of data from one place to another through electromagnetic waves. It is a mode of communication that uses free space instead of wires. Hence the data travels in the air as same as light does. Wireless communication is mostly related to radio, microwave and infrared waves.

This type of communication is quite swift with a better output. Data can be exchanged in less time. People far away from each other can easily communicate at any time e.g., use of online chatting, cell phones, e-mails etc. It has many other advantages like to install the wireless system in a building will be easy comparative to fix all wires in the building for the wired network would be time taking, complicated and difficult.

### *1.2 Wireless Network*

The system that enables wireless data communication is called the wireless network, e.g. radio channel network, TV network etc. It consists of either computers, laptops , notebooks, routers, switches, cell phones, portable phones, PDA's, related operating systems / softwares, access points (AP), base stations (BS), antennas or towers etc. One network can interconnect with other network or sub network. As WLAN [1] is one network but it can interconnect to Bluetooth [1] wireless system or can also support the wireless ad-hoc network.

## II. Background

Medium Access Control (MAC) algorithms are used to allow several users simultaneously to share a common medium of communication in order to gain maximum of channel utilization with minimum of interference and collisions. MAC is similar to traffic regulations in the highway. Several vehicles cross the same road at a time but rules required to avoid collision e.g., follow the traffic lights, building the flyovers etc. [1].

MAC belongs to layer 2, the Data Link Control layer (DLC) of the ISO OSI reference model. Layer 2 is subdivided into the MAC layer 2a, and logical link control (LLC) layer2b. The task of DLC is to establish a reliable point-to-point or point–to-multipoint connection between different devices over wired or wireless medium.

### *2.1 Basic MAC Algorithms*

Many MAC algorithms and protocols have been successfully used in wired networks for a long time. Some of them are quite famous and elegant algorithms such as ALOHA and Carrier Sense Multiple Access (CSMA). These are very basic schemes for multiple access channels, and they are also the basis for wireless channel allocation schemes.

#### *2.1.1 ALOHA*

In 1970s Norman Abramson proposed a new and reliable algorithm to solve the channel allocation problem in wired network. Abramson worked with his colleagues at the University of Hawaii to develop this method called ALOHA or Pure ALOHA. Its another version is called Slotted ALOHA [2].

Pure ALOHA is a random access protocol. A user can access the channel whenever it has data to be transmitted. Definitely, there will be a collision. However, after transmission the user waits for an acknowledgment from separate feedback channel. If there is collision, the sender waits for a random amount of time and retransmits the data. Pure ALOHA does not relate to time synchronization.

Slotted ALOHA divides the time into equal time slots of length greater than the packet duration. Each user has synchronized clock and transmits the data only at the beginning of new time slot. This helps in a discrete distribution of accessing the channel. But collision is not prevented absolutely; there is a collision with portions of data packets.

#### *2.1.2 CSMA and it's variants*

ALOHA does not listen to the channel before transmission. On the other hand, carrier sense multiple access (CSMA) algorithm is based on the concept that each station on the network is able to sense the channel before transmitting the data packet. Sensing the channel means to monitor the status of channel whether it is idle or busy. If the channel is idle/free, then station can transmit the data. But if the channel is sensed busy, the station will wait and keep on sensing the carrier till it becomes free. This method decreases the probability of collision.
There are several versions of CSMA exist:

*Non-persistent:* In this type of CSMA, a station senses the channel first. If the channel is free then it starts transmission immediately. But if channel is busy then the station does not continuously sense the channel, rather it waits for a random amount of time and then repeats the algorithm [1].

*p-persistent:* It is applied to slotted channel. Here stations also sense the medium. If the medium is free, a station transmits the packet with a probability of p, or with probability of 1-p if the station defers to next slot.

*1-persistent:* When a station wants to send the data, it first senses to the channel whether it is free or busy at the moment. If it is busy, the station waits until it becomes free. And if the station detects an idle channel, it transmits a data frame. When the channel becomes free the two or more neighboring stations can transmit data at the same time. This will cause collisions. If the collision occurs, the station waits a random amount of time and repeats the method. The algorithm is called 1-persistent because the station transmits with a probability of 1 whenever it finds an idle channel.

### *2.2 Back-off Algorithms*

Collision and loss of packets are the major problems in wireless networks compared to wired networks. Then how much time should be spent for waiting when the carrier is busy, waiting after collision or loss of packets etc. are other critical issues in wireless domain. However, some techniques and methods have also been applied besides the MAC algorithms to overcome these issues. The terminologies like random amount of time / random back-off time have been mentioned in ALOHA, CSMA and will be used in subsequent protocols too. The purpose of these techniques is to make a transparent and justified way of accessing the wireless medium. The real algorithms producing the random amount of time are the Back-off Algorithms. There are two types of such algorithms:

#### *2.2.1 Random Back-off time / Binary Exponential Back-off*

This is the mostly used algorithm [1] in order to select the random amount for the duration of waiting time in the network. Here the random amount of time is the random back-off time that counts downwards to zero. This time delays the access of medium in order to provide transparent and collision free environment for all nodes in the network. Whenever, if any node finds the medium busy in the network, it is supposed to get a random value within a **contention window** for back-off time. The node starts counting down its back-off time only when the medium becomes free. Each node may have different or same amount of time but within contention window. This random waiting time avoids collisions; otherwise all nodes would have accessed the idle medium at the same time. After finishing that random time, they start sensing the medium. As soon as a node senses the channel is busy, it loses this turn and it will select another back-off time for the next cycle. On the other hand, if a node gets the medium free after waiting for random time, it can access the medium immediately [1]. Contention window (CW) is set with an initial size e.g. CWmin = 7. The back-off time is selected from the CW and it could be any value between 1 and 7. CW becomes double + 1 at each time for every collision or lost frame. The window can take on the values 7, 15, 31, 63, 127, 255 and so on. Let maximum size of CW in this example is CWmax=255. The collision indicates the load on the network, and then doubling the value of CW can minimize the chances of collision. It is hard to select the same random back-off time using large CW. This algorithm is also called the Binary Exponential Back-off (**BEB**), because CW doubles (having linear graph) at each time of collision [1]. The value of CW is reset to its original minimum value (CW=7) as soon as any transmission completes successfully after the occurrence of collision.

## III. PROPOSED METHOD

In a wireless network, simultaneous packet transmission by nearby nodes is often undesirable. This is because any resulting collision between these packets may cause a receiving node to fail to receive some or all of these packets. This is a physical problem, which occurs before packets can be inserted into the receiver queue. Depending on the characteristics of the medium access control and other lower layer mechanisms, in particular whether retransmission of unacknowledged packets is supported, this may cause at best increased delay, and at worst complete packet loss. In some instances, these problems can be solved in these lower layers, but in other instances, some help at the network and higher layers is necessary.

The problems of simultaneous packet transmissions are amplified if any of the following features are present in a protocol:

*Regularly scheduled messages* - If two nodes generate packets containing regularly scheduled messages of the same type at the same time, and if, as is typical, they are using the same message interval, all further transmissions of these messages will thus also be at the same time. Note that the following mechanisms may make this a likely occurrence.

*Event-triggered messages* - If nodes respond to changes in their circumstances, in particular changes in their neighborhood, with an immediate message generation and transmission, then two nearby nodes that respond to the same change will transmit messages simultaneously.

*Schedule reset* - When a node sends an event-triggered message of a type that is usually regularly scheduled, and then there is no apparent reason why it should not restart its corresponding message schedule. This may result in nodes responding to the same change also sending future messages simultaneously.

*Forwarding* - If nodes forward messages they receive from other nodes, then nearby nodes will commonly receive and forward the same message. If forwarding is performed immediately, then the resulting packet transmissions may interfere with each other.

A possible solution to these problems is to employ jitter, a deliberate random variation in timing. Such jitter is employed in which transmission intervals for regularly scheduled messages are reduced by a small, bounded and random amount in order to desynchronize transmitters and thereby avoid overloading the transmission medium as well as receivers.

**International Journal of Electronics Communication and Computer Engineering**
**Volume 4, Issue 2, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209**

We extend the concept of a fixed jitter value and study the effect of varying the jitter by 10 % to 300% of the original value. We extend the implement in ns-2 for our study. In the next section, we discuss the simulation setup.

## IV. Simulation in NS2

NS2, Network Simulator 2, is an event driven network simulator launched at the University of California at Berkeley. It supports different networking protocols like TCP, UDP, queuing management, and some traffic sourcing like Telnet, CBR, and FTP. NS2 is written in C++ and OTCL, Object Oriented Tool Command Language which was developed at Massachusets Institute of Technology [8].

Evaluating result of jitter value ranging from 10% to 300% of the standard for two cases

*Case 1:* When node0 and node1 communicating in cross-connection between node2 and node3 communicating. Referred to as Cross Traffic.

*Case 2:* When only node0 and node1 communicating. Disabling connection between node 2 and 3. By commenting line $ns at 0.2 "$ftp1 start" of s3.tcl script.

Steps to be done to run the script:
Step1: Start
Step1: Create the tcl script ie s3.tcl
Step2: To run the script for 10% of standard value i.e. ns s3.tcl 0.1(Cross Traffic). Enable the code in mac-simple.cc as double jitter= Random::random()%40*100/ bandwidth_ * .1;
Step3: Recompile ns-2 by using ~/ns-2.34/make
Step4: Run the script.\
Step5: Analyze the data captured.

In the figure below we show the screen shot of simulation setup.

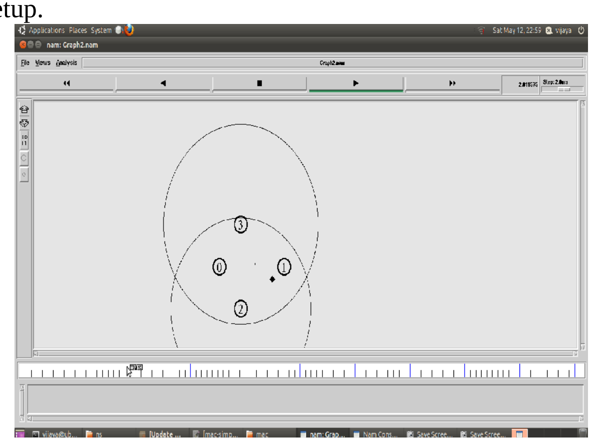

Fig.1. Setup for Single Source and Cross Traffic Simulations

In the next section, we analyze the simulation results.

## V. Result

Below table 1 and table 2, we have shown the data captured from the simulations. The values are as follows: In table 1, column 1, the jitter values are shown, in column 2 average waiting time captured in cross traffic w.r.t. each jitter value of column 1, in column 3 average waiting time captured in single source w.r.t. each jitter value of column 1.In table 2, column 2, packets received in cross traffic w.r.t. each jitter value of column 1 is shown and in column 2, packets received in single source simulation w.r.t. each jitter value of column 1, is shown.

Table 1: Case 1 (Data captured from the simulations)

| Jitter Value | Avg. Waiting Time | Received Packets |
|---|---|---|
| 0.1 | 9.30E-05 | 163 |
| 0.3 | 0.000288 | 495 |
| 0.5 | 0.000486 | 1125 |
| 0.7 | 0.000678 | 1599 |
| 0.9 | 0.000877 | 1237 |
| 1.1 | 0.001066 | 1591 |
| 1.2 | 0.00168 | 1542 |
| 1.5 | 0.00146 | 1107 |
| 2 | 0.001937 | 1534 |
| 3 | 0.002922 | 1120 |

Table 2: Case 2 (Data captured from the simulations)

| Jitter Value | Avg. Waiting Time | Received Packets |
|---|---|---|
| 0.1 | 9.09E-05 | 52 |
| 0.3 | 0.000277 | 132 |
| 0.5 | 0.000487 | 1025 |
| 0.7 | 0.000683 | 1307 |
| 0.9 | 0.000877 | 637 |
| 1.1 | 0.001066 | 1429 |
| 1.2 | 0.001166 | 1346 |
| 1.5 | 0.001458 | 1147 |
| 2 | 0.001938 | 1420 |
| 3 | 0.002909 | 1357 |

We plot the data values in the graphs below.

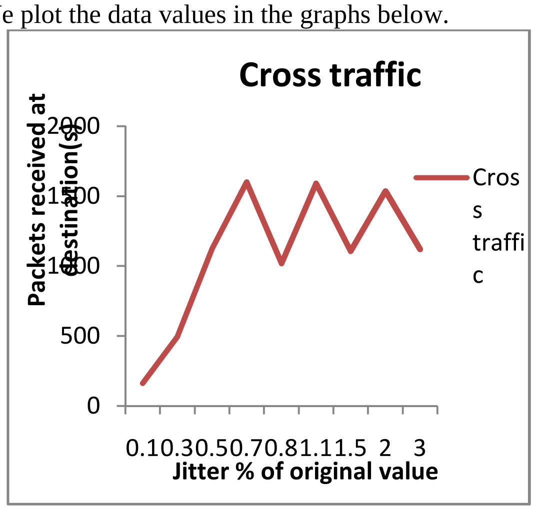


Fig.2. Total packets received at both the destinations in case of the cross traffic simulation

Figure 2 shows the total packets received at both the destinations in case of the cross traffic simulation, with increasing value of jitter. Similarly, Figure 3 shows the result for Single source transmission. Figure 4 shows the combined results of these plots.

Figure 5 shows the average jitter value as the percentage of jitter increment is changed.

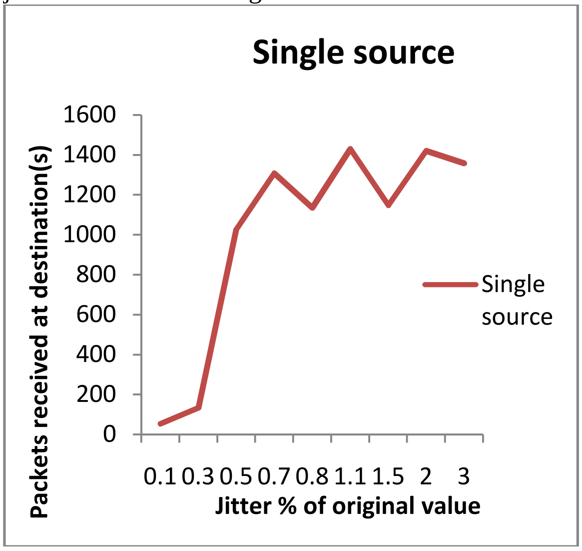


Fig.3. Total packets received at the single destination in case of the single source simulation

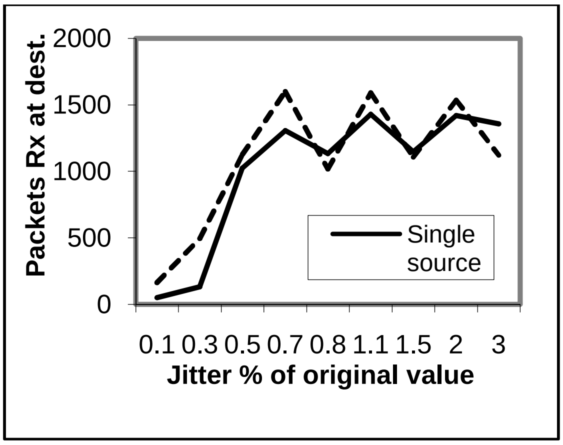


Fig.4. Combined result of single source and cross traffic

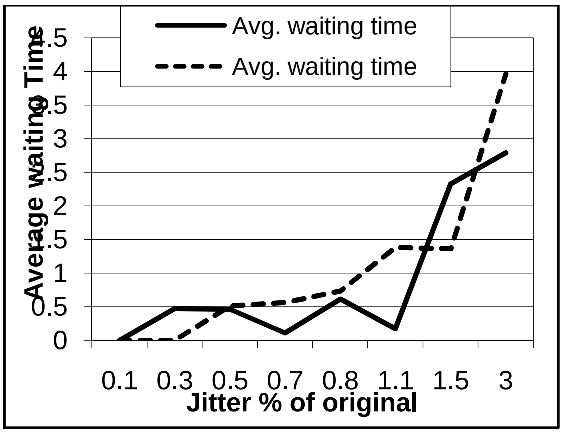


Fig.5. The average jitter value as the percentage of jitter increment is changed

## VI. Conclusion

The performance of the protocols that randomly choose the slot at which transmission occurs is bounded by a fundamental trade-off. If the contenders aggressively transmit, the probability of collision is high.

If the contenders use low transmission probability i.e. separates their transmission attempts by a large number of slots, the performance suffers because most of the slots remain empty. Although the transmission probability can be optimized, the resultant efficiency is still far from satisfactory. A conceptual change in the protocol is required to overcome the aforementioned fundamental bound.

Randomness is of paramount importance for resolving collisions. After a collision, it is desired that the implicated parts backoff for a different number of slots, in order to prevent that they collide in their next transmission attempt. Given the facts that random selection of the transmission slot limits the performance and that randomness is necessary to resolve collisions.

We evaluated the behavior of Simple MAC which uses random jitter values to give an opportunity to nodes to transmit in case of a decentralized ad hoc network. Our simulations with 4 nodes show that in case of cross traffic an increased jitter is helpful in improving collision free packet transmissions. These simulations could be extended further to determine heuristic values for jitter setting.

Further evaluations and experimentation is needed to study the effect of jitter in case of large number of nodes creating cross traffic and with different traffic patterns. This will help in determining appropriate jitter values.

## References

[1] Jochen Schiller, “Wireless LAN,” in Mobile communications, second ed., pearson publications, 2003, ch. 7, 7.3.4, pp.214–218.

[2] S.Keshav, ”Scheduling”, An enggineering approach computer Networking, Pearson publications, 2008,ch.9,pp-266-272.

[3] Yu Zheng, Kejie Lu, Dapeng Wu, and Yuguang Fang, “Performance Analysis of IEEE 802.11 DCF in Imperfect Channels," Sep 2006.

[4] Nah-Oak Song, Byung-Jae Kwak, Leonard E. Miller, ”Analysis of EIED Backoff Algorithm for the IEEE 802.11 DCF”, IEEE 2005.

[5] G. Bianchi, “Performance Analysis of the IEEE802.11 Distributed Coordination Function,” *IEEE Journal on Selected Areas in Communications*, Vol.18, No.3, PP.535-547 Mar. 2000.

[6] IEEE 802 Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, IEEE Std., 1999.

[7] Hongqiang Zhai, Jianfeng Wang, Xiang Chen and Yuguang Fang,” Medium access control in mobile ad hoc networks: challenges and solutions”, Wireless Communications And Mobile Computing, 2006; 6:151–170 Published online in Wiley Inter Science.

[8] Ns2manualhttp://www.isi.edu/nsnam/ns

[9] P. Chatzimisios, V. Vitsas, A. C. Boucouvalas and M. Tsoulfa., ‘Achieving performance enhancement in IEEE 802.11 WLANs by using the DIDD backoff mechanism,", 2007.

[10] Younggoo Kwon, Yuguang Fang and Haniph Latchman,” A Novel MAC Protocol with Fast Collision Resolution for Wireless LANs”, 0-7803-7753-2/03, 2003 IEEE.

[11] Yi-Hua Zhu, Xian-Zhong Tian, and Jun Zheng,” Performance Analysis of the Binary Exponential Backoff Algorithm for IEEE 802.11 BasedMobile Ad Hoc Networks”,978-1-61284-233-2/11,2011 IEEE.

International Journal of Electronics Communication and Computer Engineering
Volume 4, Issue 2, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

## AUTHOR'S PROFILE

**Vijaya Sagvekar**
has received M. Tech in Information Technology from NMIMS university in 2012.She has a teaching experience of more than 7 years. She is Lecturer in Computer Department in SVPP, Mumbai. Her area of interest includes Computer networking, Network Security Network Simulators.
Email: vijayasagvekar@gmail.com

**Vidya Sagvekar**
has received M.E. in Electronics and Telecommunication Engineering from Mumbai University in 2013.She has a teaching experience of more than 4 years. She is Assistant Professor in Electronics Department in KJSIEIT, Mumbai University. Her area of interest includes Computer Networking and Network Simulators. Email: vidyarsagvekar@gmail.com