

# NS2 inconsistencies of IEEE 802.11 DCF and a non-randomizing back-off in MAC

Nihad S.

**Abstract** – This work, although is regarding the implementation of a modified back-off mechanism in wireless networks, has a number of observations done in the code execution; i.e. some interesting facts from the view point of the program that executes the functioning of the aforementioned back-off mechanism.

**Index Terms**— 802.11, Ad hoc network, Contending nodes, Fairness, Idle slots, Random back-off, Throughput

## 1 INTRODUCTION

Before going into the programming aspects of the protocol modification to implement the new back-off mechanism, this section gives an insight about the wireless standard upon which the changes will be made for simulation purposes. The IEEE 802.11 standards are the first international standards for WLANs, providing general MAC layer and physical (PHY) layer specifications. At the MAC layer, IEEE 802.11 specifies a basic distributed coordination function (DCF), which is a contention based decentralized protocol utilizing the so called carrier sense multiple access with collision avoidance (CSMA/CA). The major challenges faced MAC protocol for wireless ad hoc networks are due to the dynamic nature of the topology and the shared wireless media. There are serious performance degradations under heavy contention and these demerits are present in the DCF of IEEE 802.11.

## 2 BACK-OFF IN DCF

In the DCF, sending stations stand in contention to each other. They use the medium access scheme carrier sense multiple access with collision avoidance (CSMA/CA). Before sending data, the station has to sense the channel. If the medium is at least idle for DIFS the station is allowed to send. In the case of a busy medium the station starts the random back-off procedure. It determines a slotted random back-off time which is a product between duration of a slot and Random, where Random is a pseudo-random integer value out of the uniformly distributed contention window  $[0, CW]$ . If the medium is idle again at least for DIFS, it decrements the random back-off time as long as the medium is idle. The station is allowed to send immediately if the random back-off time equals zero.

The random back-off procedure has to be started after every transmission. In the case of a successful acknowledged transmission the procedure will be started after the received acknowledgment (ACK). Otherwise the procedure will be started after the expiration of the ACK timeout interval.

A collision occurs if two (or more) stations have detected the medium as idle for DIFS, both are allowed to send and both start their transmissions immediately. This affects the random back-off procedure after the expiration of the ACK-

timeout interval.

## 3 NEW BACK-OFF MECHANISM

The binary exponential back-off mechanism is one of the basic elements that constitute the IEEE 802.11 protocol. The models of the back-off mechanism have been developed with the assumption that collisions occur only due to nodes within the carrier sensing range and the collision probability is constant in steady-state. However, the transmission collisions can occur due to hidden nodes and these tend to occur consecutively, contrary to the collisions due to nodes within the carrier sensing range. Consecutive collisions increase the back-off time exponentially, resulting in less frequent transmission attempts. Ignoring this collision characteristic in modeling the back-off mechanism can produce large errors in the performance analysis of networks.

The inspiration for the paper came with the basic thought of doing the opposite of the existing method, which is to 'non-randomize' the back-off procedure. In simple words, the idea is to multiply the contention window with a non-negative, non-zero valued constant, in the scenario of consecutive collisions occurring in the network for a node. This constant value remains fixed for the duration of a simulation.

Two nodes are considered here for the simplicity of explanation. In the scenario when both of them transmit together, a collision occurs, obviously. The proposed protocol mirrors CSMA/CA functioning for like an 'initialization' period. A random back-off occurs the first time a collision occurs between the nodes. After both of them have successfully transmitted data, each of them are assigned constant back-offs. In simple words, if the random back-off for one node is 'x' and for the other node is 'y', and a constant 'a' is applied to them after successful transmission; i.e their respective next back-offs being 'x+a' and 'y+a', obviously, these two nodes would not interfere with each other again. Thus, once two nodes have transmitted successfully, its back-offs need not be random any more and by making it a constant, it will be ensured that the collision process does not occur again. The scenario is applicable for larger number of nodes as well.

## 4 IMPLEMENTATION DETAILS

The network was simulated on NS2 (Network Simulator version 2.36) platform. Much of the work has been done in modifying the 'backend' code of the simulation tool with major changes being made in source code of 802.11, namely - mac-802\_11 (.cc, .h) and mac-timers (.cc, .h)

The protocol is also applied in ad hoc networks and hence none of the functions related to infrastructure mode were used or modified.

The main process is of incorporating a constant value into the contention window or the random number range. In other words, the procedure constitutes of making either the contention window or the random number obtained from a specific range to be a multiple of the applied constant, based on the collision performance. The initial task is to find out whether or not the node is experiencing consecutive collisions at the point of time of back-off. The process of finding whether consecutive collisions are taking place, was implemented in mac-802\_11.cc file; it was the only major modification in the code to the file. This was implemented by a simple procedure of setting up a counter that returns a non-zero value if the condition of collision (`rx_state_ == MAC_COLL`) is true for a node. The function returned either a zero (indicating no collisions) or a non-zero value (specifying successive collisions taking place in the node). This value is carried over to the mac-timers file where the procedure for implementing a constant back-off is put into practice.

Two distinctive course of action takes place based on whether there are successive collisions or not. If there are repeated collisions taking place, the contention window is modified with respect to the assigned constant.

If there are no collisions, the range over which random variable is selected is made to be a multiple of the constant, along with the contention window value set to the minimum.

Making the threshold for packets considered to be of RTS to 3000 bytes effectively disables the RTS/CTS mechanism, which is not used in this protocol. The setting in the file called ns-default.tcl (which stores the initial values of the protocol parameters) is also altered in order to assist the protocol's method in preventing the RTS/CTS course of action.

The setting: `Mac/802_11 set bugFix_timer_ true;` is considered to be a fix for the scenario when RTS/CTS not used. The setting indicates about a solution to a bug that was present in the previous version of the simulation tool. It facilitates a sub case where the working of defer time function is avoided. It has been of wide criticism that the use of defer time in the implementation of 802.11 deviates from the actual process in the protocol.

The protocol is also applied in ad hoc networks and hence none of the functions related to infrastructure mode were used or modified.

The paper considers simple ad hoc network with every node at the transmission range of each other, along with 'disabling' RTS/CTS mechanism. One of the reasons RTS/CTS is not used is the fact that it adds a certain amount of overhead. Also, if the problem of hidden terminals is avoided by making all the nodes in the network function at a transmission range,

the requirement of RTS/CTS does become trivial.

One of the major areas of concern in legacy 802.11 DCF is maintaining constant high throughput as well as fairness, especially when the number of contending nodes increases.

Two sets of experiments were done to come to a conclusion based on the above proposed mechanism of back-off implemented in the MAC protocol.

Starting with lesser number of nodes, as far as 50 simultaneous transmissions were employed between nodes to examine the variation of throughput. This was done in two methods:

- (i) Employing all source nodes transmitting to a single sink node, and
- (ii) Each source node, transmitting at the same time to their respective sink node. (i.e each source node having a corresponding sink node)

### *Experimental setup*

A 500 x 500 m<sup>2</sup> unit with CBR (constant bit rate) traffic sources were used. The environment was static ad hoc with the absence of hidden terminals.

The network conditions used were light in nature with maximum of 50 nodes employed in the network.

The experiment was done such that each source node were timed to transmit at a particular point of instant of simulation to their respective sink node.

Maximum and minimum value of contention window is taken to be 1 and 1024. Slot duration is 16  $\mu$ sec.

The output values from the simulator were obtained in trace files, specifically recording each microsecond's proceedings of the simulation. With the help of awk file, throughput parameter were calculated and verified with the theoretical values and they were plotted using Matlab software.

## 5 RESULTS AND INCONSISTENCIES IN NS2

There was a number of interesting outcomes as a consequence of the constant back-off implementation. The performance clearly depended on the value of the constant and the number of contending nodes in the network. The feat mirrored the fact that the throughput remains high and constant (around 6.1 Mb/s) as long as the value of the constant is less than or equal to the number of contending nodes. The performance is compared along with that of legacy DCF to get a good idea about the variations. It can be seen that the throughput behavior varies undesirably as the number of nodes increases in legacy DCF. The proposed protocol's behavior, on the other hand, remain constant irrelevant of the number of contending nodes. Initially, when the experiment starts, the protocol takes a few seconds for the 'initialization' period. Its throughput does not begin with the highest values and this can be reasoned due to its 'settling down' time, that occurs due to CSMA/CA traits present in the protocol. To

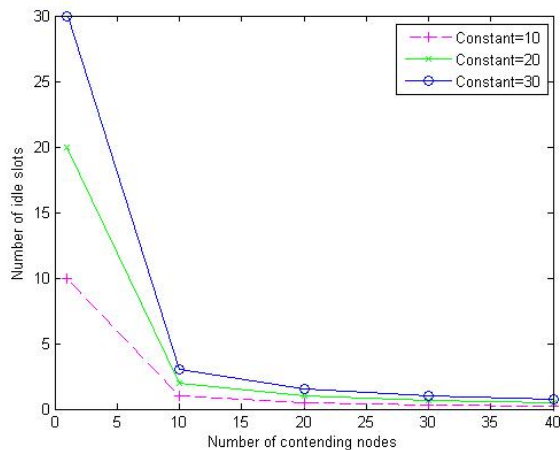


FIGURE 1: IDLE SLOTS VARIATION BASED ON THE CONSTANT AND NUMBER OF NODES (RGB file)

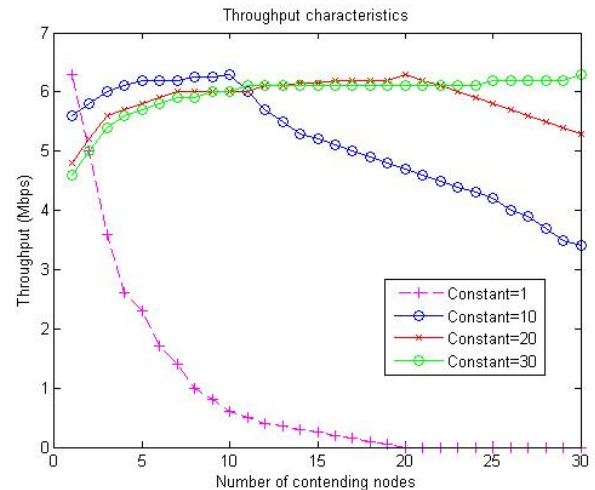


FIGURE 2: THROUGHPUT VARIATION UNDER DIFFERENT CONSTANTS AND COMPETING NODES (RGB file)

sum up the throughput performance, for DCF, as the number of actively communicating nodes in the network increases, throughput is never a constant value and importantly, does not remain high either. The problem of packet drop/loss popping up as soon as the number of parallel transmissions or the number of competing nodes increased in legacy DCF was not surprising. This was fairly straightforward reason for the throughput varying the way it did in DCF.

A large number of researches have used [4] to calculate or compute the fairness performance of the network, from the throughput values. This paper has also used the aforementioned work to figure the fairness index performance. It can be seen that, for the new protocol, due to throughput being consistent, the fairness index closes on unity, which is the ideal value.

$$\text{Fairness index} = ((\sum_i X_i)^2) / (N * \sum_i (X_i)^2) \quad (1)$$

Where N number of nodes and  $X_i$  denotes throughput of  $i$ th node. The feature implies that every node in the network gets a fair chance at transmission.

Throughput value starts to decrease as soon as the number of nodes actively participating in transmission goes beyond the value of the constant. It is also interesting to note the effect or impact of idle slots on the overall performance. This paper has taken an assumption of the ratio of the constant to the number of contending nodes to be an indication of the number of idle slots in the channel. This value is instrumental in the throughput analysis, because, as the ratio becomes a fraction (value becoming less than unity), the channel experiences 'suffocation' with more number of nodes contending for a comparatively lesser value of contention window, resulting in collisions and thereby decreasing the throughput. Some inferences drawn from this ratio can be summarized as:

As the number of nodes increase, the value of the above mentioned ratio will decrease and it will reach unity at a point, when the network performance will be at its peak.

Thereafter, as number of nodes becomes higher than the constant value, the network will not have enough number of slots to accommodate all users fairly and collisions may result. Thus, the performance will start to diminish.

Figures 1 and 2 demonstrate the effects mentioned above. In figure 1, three scenarios are generated with constant values 10, 20 and 30. As soon as these values, with respect to the number of nodes competing for transmission goes below 1, the channel will not accommodate all the nodes fairly and performance degradation follows. The predicament in terms of throughput execution can be seen in figure 2.

Having extensively worked in NS2 simulation tool, it has to be mentioned that there are some inconsistencies in the source code for the 802.11 implementation. For instance, consider the following statement, which appears in a number of places, indicating the necessity of starting the back-off timer at that instance. 'If the medium is NOT IDLE, then we start the back-off timer.' Referred from various papers and texts, it is the author's belief that the back-off timer addressed above will start (or resume) once the medium is idle, with no transmissions taking place at all. The logic behind the above statement was not clear, although it did not affect this paper's work.

The implementation of 'defer timer' also produced unwanted complications. The IEEE 802.11 DCF already has a back-off timer and the introduction of this deferring timer results in somewhat 'doubling' the effect of back-off. This, in turn, results in the deviation of the whole back-off procedure of DCF.

The finding of random variable within a range also has some drawbacks, especially when the minimum value of the range is a non-integer value. Current implementation addresses the case of the range always starting from zero.

These were rectified accordingly to prevent any hindrance to the implementation of the new back-off mechanism and this paper's work.

## 6 CONCLUSION AND FUTURE WORK

The 802.11 DCF is a simple random access method with an important protocol overhead that works fairly well for a small number of contending stations. This becomes increasingly challenging as the number of stations increase.

Although the modified MAC protocol described above produces desirable results, there are number of pre-requisites that hinder the 'completeness' of the work. The future work will include modification and developing methods that will result smooth operation of the above protocol in networks that may or may not have the hidden terminal problem. Also, the requirement of changing the value of the constant with respect to the number of nodes, which in turn is the backbone of the whole process, has to be negated and a suitable algorithm has to be developed so that the protocol can automatically modify the constant, as the number of competing nodes in a network increase. The protocol is considered to be effective for other wireless standards as well (including 802.11g and 802.11n standards) and its investigation and analysis will also constitute the next step of this work.

### REFERENCES

- [1] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function", *IEEE Journal on Selected Area in Communications*, vol.18, no.3, pp. 535-547, 2000.
- [2] F. Cali, M. Conti and E. Gregori, "IEEE 802.11 protocol: Design and performance evaluation of an adaptive backoff mechanism," *IEEE Journal on Selected Area in Communications*, vol. 18, pp.1774-1786, Sept. 2000.
- [3] Ke Liu, "Understanding the implementation of IEEE MAC 802.11 standard in NS-2"
- [4] R. Jain, A. Durrett, and G. Babic, "Throughput fairness index: An explanation," in *Proc. ATM Forum*, Feb. 1999, Doc. 99-0045.
- [5] Carlos M. Ram´irez, Josep Paradells Asp´as and S`onia P. Mansilla, "Analysis of Consecutive Idle Time Slots", *The 17th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, 2006.
- [6] G. Berger-Sabbatel, A. Duda, O. Gaudoin, M. Heusse, and F. Rousseau, "Fairness and its Impact on Delay in 802.11 Networks," in *Proc. of GLOBECOM 2004*, 2004.
- [7] Wanrong Yu, Jiannong Cao, Xingming Zhou, Xiaodong Wang, Keith C. C. Chan, Alvin T. S. Chan, and H. V. Leong, "A High-Throughput MAC Protocol for Wireless Ad Hoc Networks", *IEEE Transactions on Wireless Communications*, Vol. 7, pp 135-145, January 2008.
- [8] Ha Cheol Lee, "Theoretical Maximum and Saturation Throughput in The 802.11 Wireless LAN", *Third International Conference on Networking and Services*, pp 105 - 111, 2007.
- [9] "The network simulator version 2.36 - ns," in, <http://www.isi.edu/nsnam/ns>