European Journal of Scientific Research
ISSN 1450-216X Vol.45 No.2 (2010), pp.310-323

http://www.eurojournals.com/ejsr.htm

An Adaptive Contention Window Control for Improving DCF Throughput and Fairness

Ali Balador
Computer Engineering Department, Science and Research Branch
Islamic Azad University, Tehran, Iran
E-mail: Balador@ieee.org
Tel: +98-912-3302341

Mahtab Ghasemivand
Computer Engineering Department, Science and Research Branch
Islamic Azad University, Tehran, Iran
E-mail: m.ghasemivand@gmail.com
Tel: +98-912-6977277

Ali Movaghar
Department of Electrical and Computer Engineering, Sharif University of Technology, Tehran, Iran
E-mail: Movaghar@sharif.edu
Tel: +98-21-66166628

Sam Jabbehdari
Computer Engineering Department, North Tehran Branch, Islamic Azad University, Tehran, Iran
E-mail: s_jabbehdari@iau-tnb.ac.ir
Tel: +98-21-88887983

Abstract

The IEEE 802.11 provides a MAC layer protocol for controlling competition among nodes to access the channel in wireless local area network. Recent works show that this standard has not suitable performances in mobile ad-hoc networks and especially in error prone channels. Many researchers proposed many algorithms to improve this standard like HBCWC (History Based Contention Window Control) scheme has significant performances but also has fairness problem. In this paper, we present a novel contention-based protocol to improve fairness and throughput together. We use an array to keep history of network collision and based on array information, we optimize the contention window. The main point is that we get higher priorities to nodes had unsuccessful transmissions unlike most of researches. This helps us to solve fairness problem. Simulation results show that compared to the IEEE 802.11 DCF and HBCWC scheme, our algorithm has better performances in term of throughput, fairness, and network overhead load.

Keywords: Mobile Ad-Hoc Network, IEEE 802.11 DCF, Backoff Algorithm, Fairness

1. Introduction

Mobile Ad-Hoc Networks (MANETs) have achieved a large amount of growth in recent years. The IEEE 802.11 access protocol which was originally designed for wireless local area networks has been invoked repeatedly in the context of MANETs to provide better performances in these networks.

In general in such scenarios, wireless nodes have a shared channel and every node should compete for the channel before it can send its own packet.

The IEEE 802.11 provides detailed Medium Access Control (MAC) and physical layer (PHY) specification for wireless networks. IEEE 802.11 MAC contains two coordination functions, namely, Point Coordination Function (PCF) and Distributed Coordination Function (DCF) which support the infrastructure and Ad-Hoc configuration. [1]

PCF depends on a central coordinator to allocate channel resource and provide services without any competition; While DCF is a mandatory and contention-based protocol. In MANETs, due to lack of access points, contention-based distributed channel access protocols are more efficient.

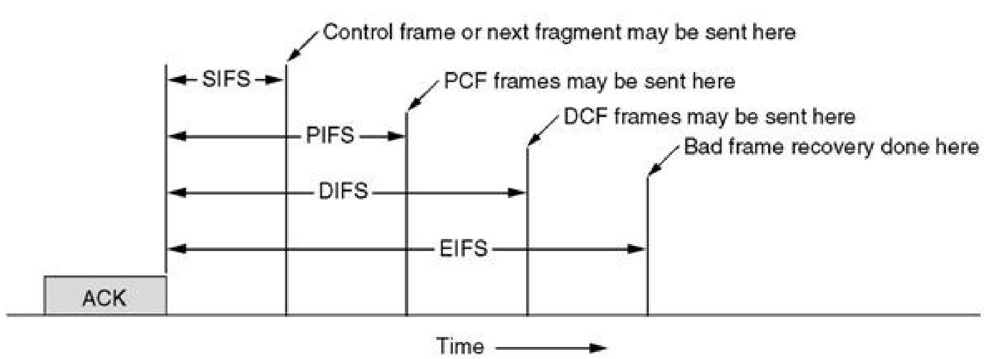
DCF is basically a Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) mechanism that gets some advantages from MACA scheme. DCF defines two channel access methods: basic mode and RTS/CTS access mode.

According to DCF, a node wishing to transmit packets senses the channel, if the channel is busy then it defers. If the channel is free for a specific time then the node is allowed to send. Before sending a data packet, a node should send a RTS (Request-To-Send) message to inform the receiver and other neighboring nodes that it wants to start a new communication. If the destination node replies it with a CTS (Clear-To-Send) message, it can start data transmission.

IEEE 802.11 uses new method for carrier sensing that called virtual carrier sensing. In this method, every node has their Network Allocate Vector (NAV) variables. When nodes send RTS or CTS packet they should calculate the transmission time and put it in control packets. When other nodes (except receiver) get these packets, they update their NAVs and don't try to access the channel during this time.

IEEE 802.11 uses this mechanism in order to reduce the probability of collision. To reduce the probability of collision, IEEE 802.11 implies other methods. It sets some Inter Frame Space (IFS) intervals that a node should waits for a predefine time before sending new packet. IEEE 802.11 defines four kinds of IFS, which is showed in Fig. 1. Shortest one is Short-IFS (SIFS) that has most priority. It uses for control packets and gets most priority to control packets such as RTS, CTS or ACK (Acknowledgement). DCF-IFS (DIFS) is the basic interval. Two other intervals are PCF-IFS (PIFS) and Extended-IFS (EIFS). PIFS is used for contention-free protocols and EIFS intervals for showing erroneous frame. [2]

Figure 1: Inter Frame Space Time Intervals



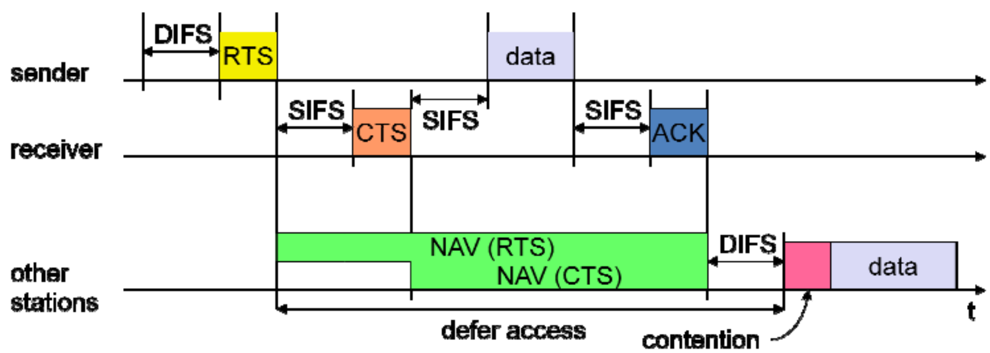
In DCF, whenever a node has a packet to transmit, it waits for a time that is involved a random backoff time and a DIFS time interval. The backoff time is generated uniformly from (0, CW-1), where CW represents the size of contention window.

As long as the carrier is sensed idle for a period of DIFS, the node starts its backoff timer and decrements its backoff value by one. When the backoff counter is reduced by one the node checks the channel if the channel is busy, the backoff counter is frozen until the next idle DIFS is sensed. When the backoff counter reaches zero, the node starts packet transmission. Other nodes that cannot access the channel double their contention window and when a node achieves successful transmission it returns its CW value to the predefined value CWmin.

When a node sends its data packet it waits for ACK packet. If sender receives ACK packet it can understand that the transmission was successfully but if it doesn't receive ACK packet it should doubles its CW value and sends the data packet again. Because nodes have one antenna they cannot listen to the channel when they send a packet.

IEEE 802.11 defines a transmission attempt limit and nodes that their transmission attempts reach this value should drop their packets. Fig. 2 shows IEEE 802.11 DCF mechanism.

Figure 2: IEEE 802.11 DCF Mechanism



2. Previous Research

Numerous research efforts have been proposed to reduce the collision probability by modifying the default BEB (Binary Exponential Backoff) algorithm of the IEEE 802.11 by novel backoff schemes or selecting an intermediate value instead of resetting the CW value to its initial value.

Based on best of our knowledge, two factors can show the network condition. One factor is the number of nodes (or active nodes) and the other factor is packet length. In previous researches, all scientists try to modify the contention window based on network level. But these schemes like [3-9] bring high overhead because of their complex computations. These schemes overhead is not acceptable in mobile communication.

To eliminate this overhead some researchers try to modify the CW value with the static scale. Important factor of this group is its lowest overhead. But the negative point is that they do not pay attention to the network load. This group involved methods such as MILD [10], DIDD [11-13], EIED [14,15] and [16, 17].

Except these methods, other schemes imply other ways like [18,19] that divide backoff range into small ranges and modify both upper bounds and lower bounds of the ranges unlike IEEE 802.11 DCF. The others like FCR [20] and LMILD [21] have planned to modify the CW value for any node that overhearing a collision.

HBCWC [22] that uses the main point of each previous group and introduced a channel status array for keeping the network history shows significant improvements in throughput and delay. But this scheme has two problems. It can be seen a lot of fluctuations in throughput and it has not good result in fairness. The fairness is reduced in this scheme.

In this paper, we proposed a novel algorithm that saves the history of network condition but we design a new algorithm in order to change backoff range in order to smooth the throughput and improve the fairness.

3. Research Method

HBCWC proposed a novel backoff mechanism in which the history of packet lost is taken into account for optimizing the CW value. A three-element array is used to save the packet lost history in the channel and the variation of this array shows eight network levels. This scheme assigns various CW value to every level. In HBCWC, if the network has better state it decreases the CW value and it increases the CW value when the network has not suitable condition.

In this study, we uses exactly opposite method for assigning network levels. It causes an increase in throughput and fairness in the same time.

3.1. Channel State Vector

In our scheme, we use the same array like HBCWC. We check the channel error upon each transmission trials (each time the node transmits the packet successfully or not). Based on this checking we update the CS array and then modify the CW value.

When a node transmits a packet if the packet lost occurs in data or ACK packet as a result of channel error or collision then the channel status bit will be set to ‘0’ and otherwise, if the packet is transmitted successfully the CS array is updated with ‘1’.

Upon each transmission trial we shift the array values. It causes the oldest one in the array to be removed and new one to be stored in the array.

3.2. Changing the Backoff Range

In this study, we check the channel like DCF but we use the opposite way in comparison to DCF for changing the backoff range. To providing better fairness, we increase the CW value when successful transmission happens. It causes an increase in fairness because it gives opportunities to others that cannot get access to the channel. This way provides fairness improvement.

Table 1 shows equations that are used to change the backoff range. In this method, we prioritize older ones comparing to new one because the new state does not show the network status well. In order to this decision, we can eliminate sudden reaction to the network changes.

Table 1: CW Modification

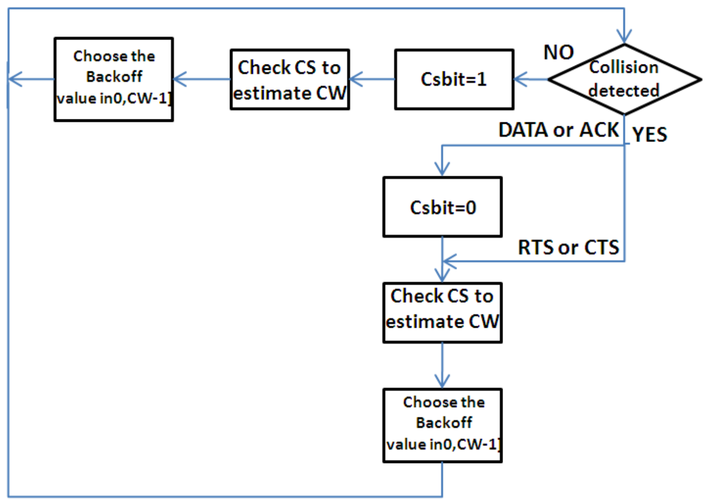
Status	CW Range
000	CW = CWmin
100	CW = CW * (X/Y)
001	CW = CW * (2 * (X/Y))
101	CW = CW * (2 * (X/Y))
010	CW = CW * (Y/X)
110	CW = CW * (2 * (X/Y))
011	CW = CW * (2 * (Y/X))
111	CW = CW * (X * Y)

Figure 3 shows the operation of this scheme. If the packet lost occurs in data or ACK packet because of channel error or collision then a channel state bit will be set to ‘0’ and CS array is updated and new backoff value is calculated. Otherwise, if the packet lost occurs in RTS or CTS packet we don’t save it in array.

In contrast, when the node transmits a packet without any error the CS array will be set to ‘1’ and the new backoff value is chosen based on the CS array.

In this scheme, we initialize CS array with 000 and we have two variables (X, Y) for assigning a new CW value to each level. We set X, Y with 1.1 and 1.9 in an initial state.

Figure 3: New Algorithm Operation



This algorithm needs extra memory space for CS array that is used for saving new variables and extra computations for five additional operations that are used to select the next CW size. But the total cost of this method is cheaper than the cost that other methods used for estimating number of nodes.

4. The Results

5.1. Simulation Model

In this section, we study the performance of our new algorithm in comparison with IEEE 802.11 DCF by using NS-2 (version 2.28). [23]

Our simulation are based on a 1000 by 1000 meter flat space and 50 wireless nodes. Simulation time was set to 600 seconds. The size of data payload is 512 bytes and each node generates data packet at the rate of 4 packets per second. The propagation range for each node is 250 meters and channel capacity is 2 Mb/s.

We utilize random waypoint model as the mobility model. The minimum speed for the simulation is 0 m/s while the maximum speed is 20 m/s. pause time is selected 50 seconds.

We present the channel error with basic error model in NS-2. We assume that error unit and error rate is respectively packet and 0.1.

5.2. Evaluating Metrics

Packet Delivery Ratio: the Packet Delivery Ratio (PDR) which represents the ratio between the number of packets originated by the application layer source and the packets received by the final destination.

Average End to End Delay: the average end to end delay which calculates the average time required to receive the packet.

Average Throughput: the average throughput which is the amount of data successfully received in a given time period that it is measured in kilo bits per second (Kbps).

Fairness Index: we use Jain's fairness index to evaluate the fairness among the flows. For a given set of flows of throughput (b_1, b2, b3,……, bn), the fairness index is defined in equation(x).

$$(\sum bi)^2/n^*(\sum bi^2) \quad (1)$$

Where n stands for number of nodes which are participating in sending the data packets in the network and b_i for the throughput of the i^{th} node. Fairness index is always between 0 and 1. A lower value implies poorer fairness. If the throughputs achieved by all the senders are same, then the farness is 1.

5.3. Simulation Results

We analyze the performance of our mechanism and the IEEE 802.11 DCF standard in error prone channel. Fig. 4 depicts average end to end delay in comparison to IEEE 802.11 DCF. It shows that we have a decrease about 35.7% in delay.

Fig. 5 illustrates the network overhead load. The network overhead load improved significantly. We can see 53.29% increases in network overhead load. Fig. 6 and Fig. 7 show the packet delivery ratio and throughput. We have 15.54% and 16.96% improvement, respectively. As we see in Fig. 8, the fairness index increased. Its improvement can be as much as 8.9%.

Figure 4: Average End to End Delay of New Algorithm vs. IEEE 802.11 DCF

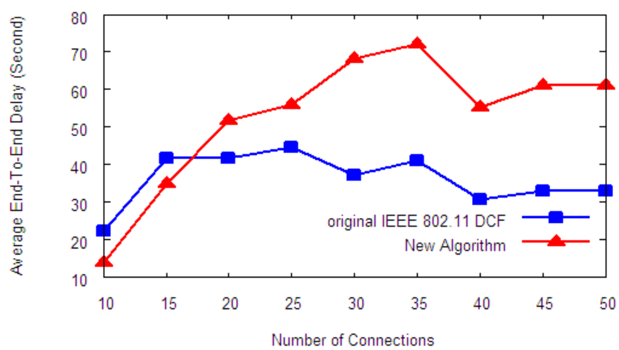


Figure 5: Network Overhead Load of New Algorithm vs. IEEE 802.11 DCF

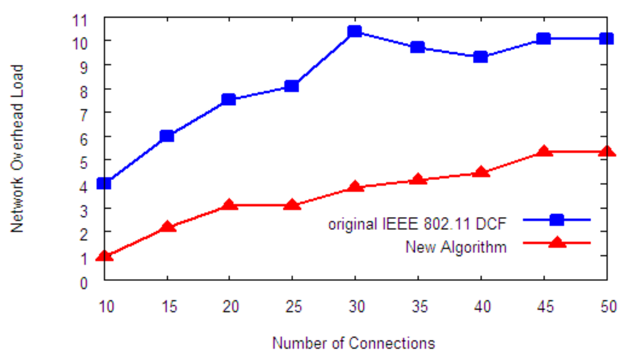


Figure 6: Packet Delivery Ratio of New Algorithm vs. IEEE 802.11 DCF

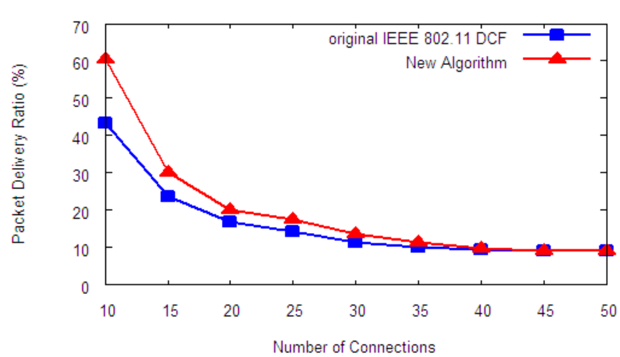


Figure 7: Throughput of New Algorithm vs. IEEE 802.11 DCF

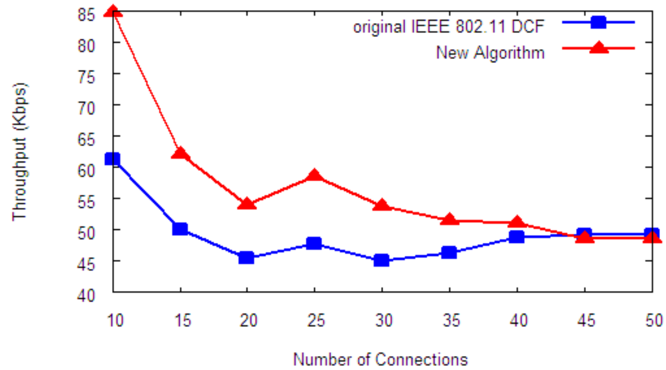


Figure 8: Fairness Index of New Algorithm vs. IEEE 802.11 DCF

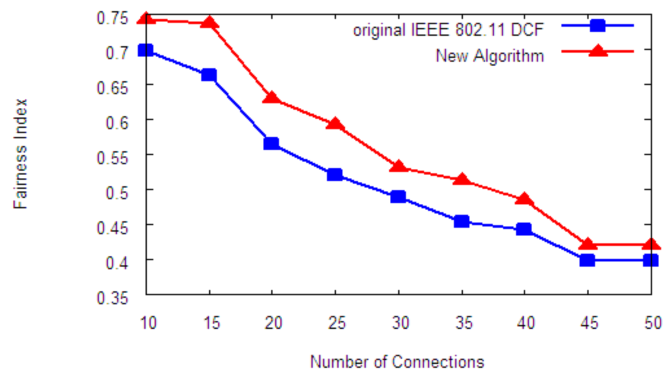


Fig. 9 to Fig. 13 shows the difference between our algorithm with HBCWC algorithm and IEEE 802.11 DCF. We can see in Fig. 9, HBCWC has better end to end delay and new algorithm has the worse one. In this method, we have a decrease in delay because we prioritize nodes that have unsuccessful transmissions.

Fig. 10 illustrates network overhead load. We have the best network overhead load comparing to the others. Fig. 11 and Fig. 12 show packet delivery ratio and throughput. In our new scheme we have an increase in comparison to IEEE 802.11 DCF but HBCWC has better improvement in some number of connections. But our improvement in new algorithm is more stable than HBCWC. These fluctuations in HBCWC cause 25.03% decrease in fairness index, that we can see in Fig. 13.

Figure 9: Comparison of Average End to End Delay in New Algorithm, IEEE 802.11 DCF and HBCWC

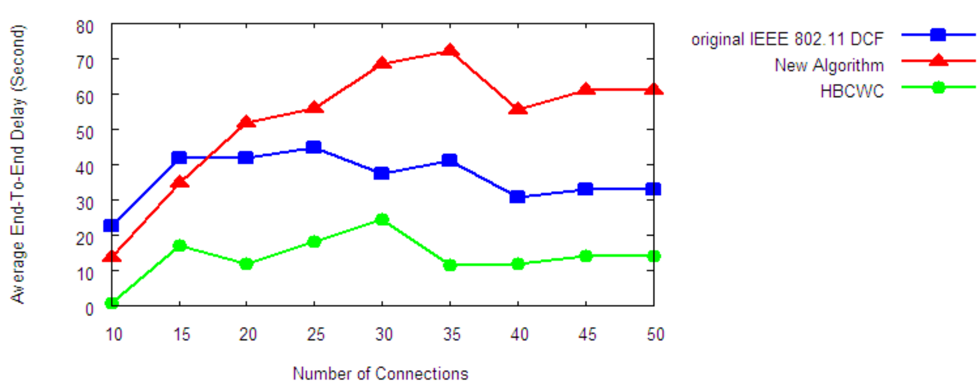


Figure 10: Comparison of Network Overhead Load in New Algorithm, IEEE 802.11 DCF and HBCWC

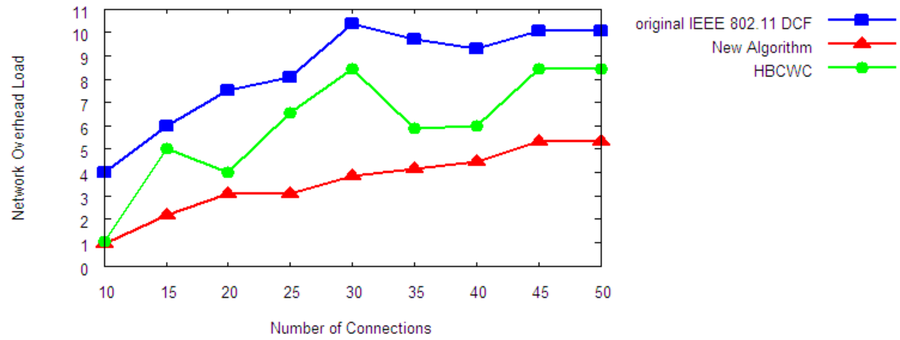


Figure 11: Comparison of Packet Delivery Ratio in New Algorithm, IEEE 802.11 DCF and HBCWC

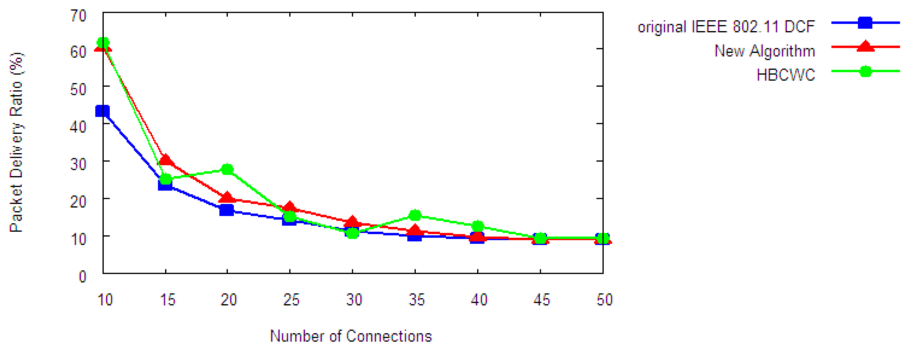


Figure 12: Comparison of Throughput in New Algorithm, IEEE 802.11 DCF and HBCWC

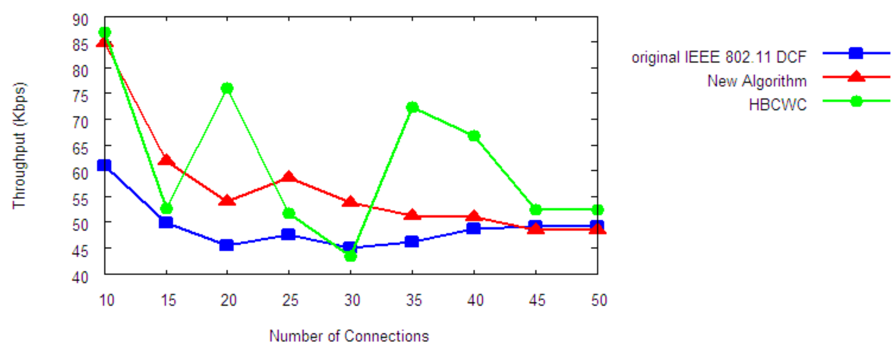
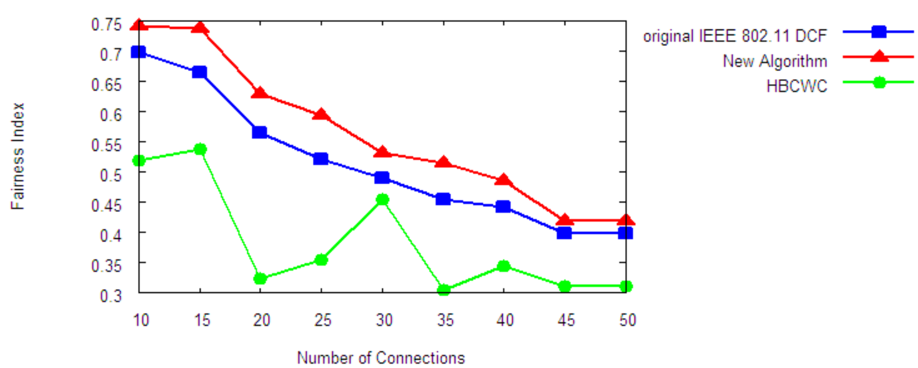


Figure 13: Comparison of Fairness Index in New Algorithm, IEEE 802.11 DCF and HBCWC



In previous figures, we found stable and significant improvements in throughput and fairness index but the negative point is decrease in end to end delay. We use two parameters x, y for calculating new CW value. Because of that optimizing these parameters are important.

Following figures show our algorithm performances per values of y. Fig. 14 to Fig. 18 show the performances when y is 1.6. Fig. 14 and Fig. 15 show improvements except 25 to 40 number of connections for y=1.6 in end to end delay and network overhead load. Fig. 16 and Fig. 17 illustrate the packet delivery ratio and throughput. Fig. 18 shows the fairness index. As shown in the figure, we have improvement in fairness.

Figure 14: Average End to End Delay of New Algorithm (y=1.6 and 1.9)

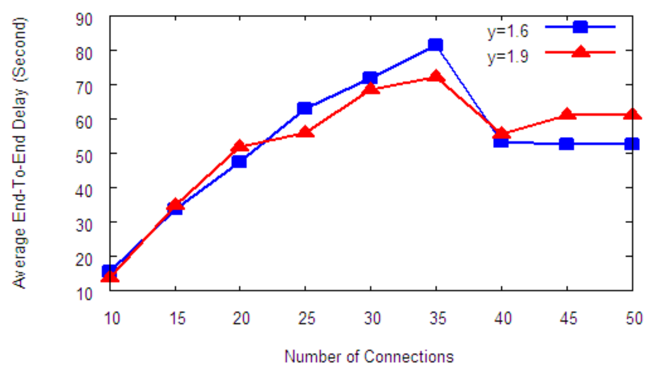


Figure 15: Network Overhead Load of New Algorithm (y=1.6 and 1.9)

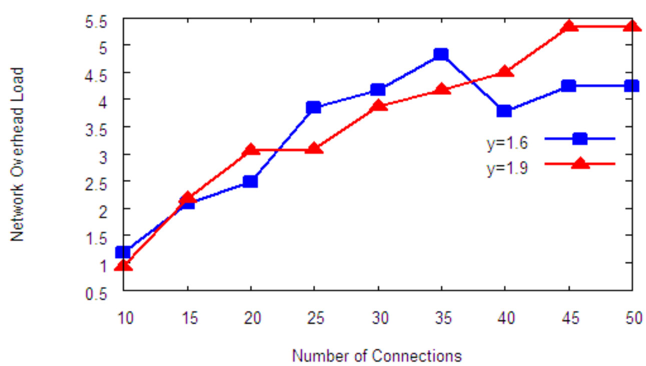


Figure 16: Packet Delivery Ratio of New Algorithm (y=1.6 and 1.9)

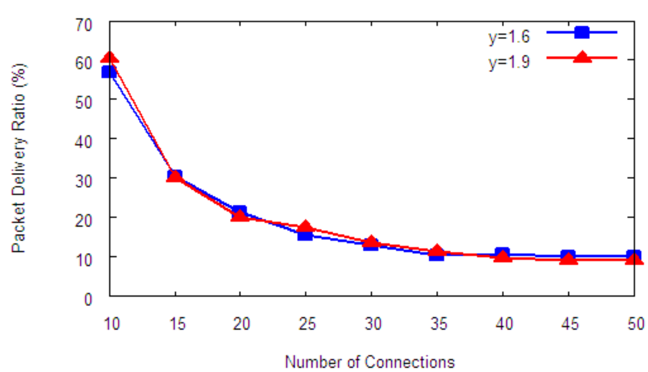


Figure 17: Throughput of New Algorithm (y=1.6 and 1.9)

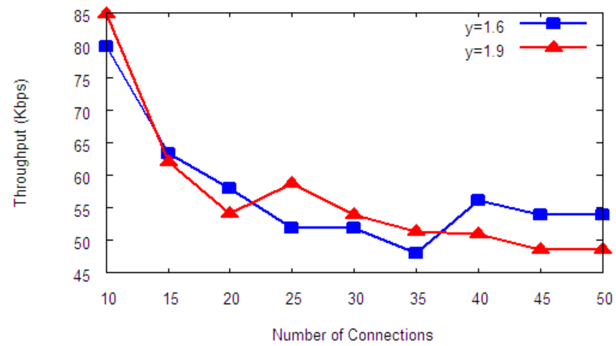


Figure 18: Fairness Index of New Algorithm (y=1.6 and 1.9)

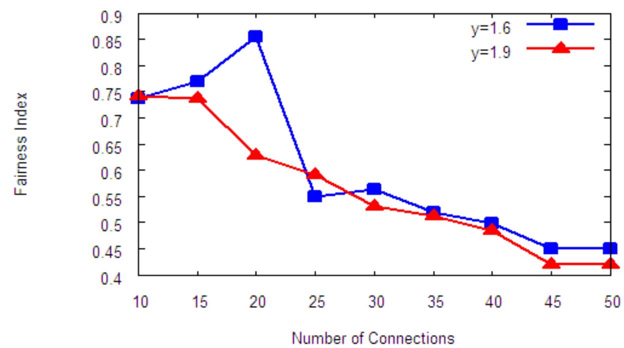


Fig. 19 to Fig. 23 show the performances for y=1.2. We can see improvement in end to end delay and network overhead load and throughput but a decrease in packet delivery ratio.

Figure 19: Average End to End Delay of New Algorithm (y=1.2 and 1.9)

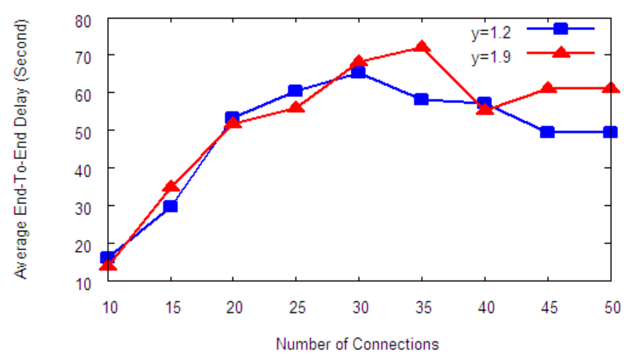


Figure 20: Network Overhead Load of New Algorithm (y=1.2 and 1.9)

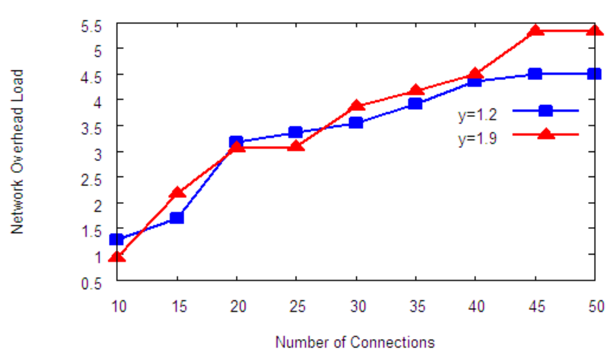


Figure 21: Packet Delivery Ratio of New Algorithm (y=1.2 and 1.9)

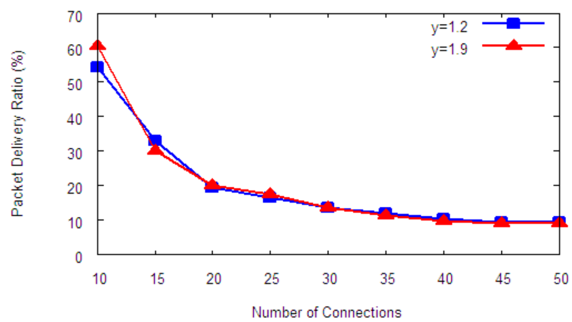


Figure 22: Throughput of New Algorithm (y=1.2 and 1.9)

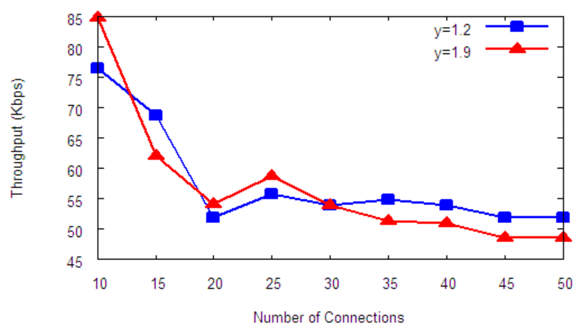


Figure 23: Fairness Index of New Algorithm (y=1.2 and 1.9)

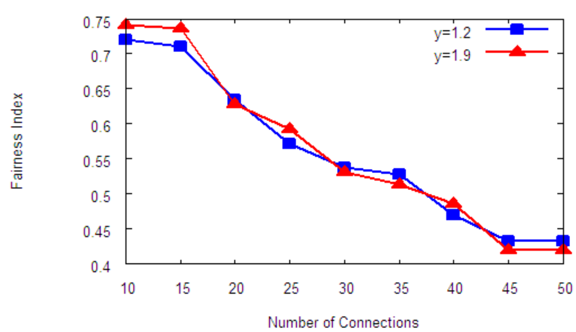


Fig. 24 to Fig. 28 illustrate the performances for y=1.4. End to end delay improved but other performances show the decrease.

Figure 24: Average End to End Delay of New Algorithm (y=1.4 and 1.9)

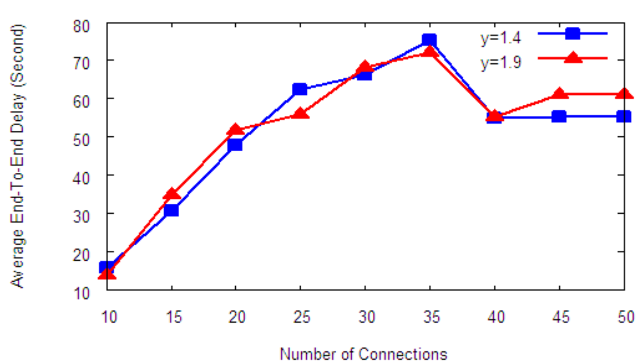


Figure 25: Network Overhead Load of New Algorithm (y=1.4 and 1.9)

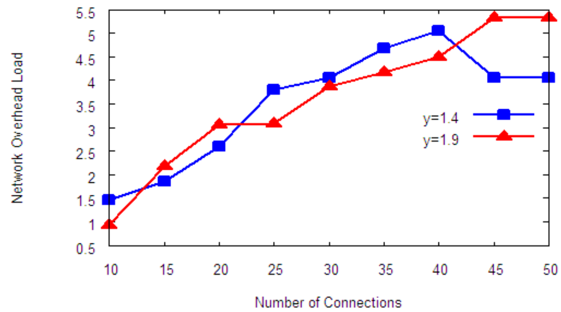


Figure 26: Packet Delivery Ratio of New Algorithm (y=1.4 and 1.9)

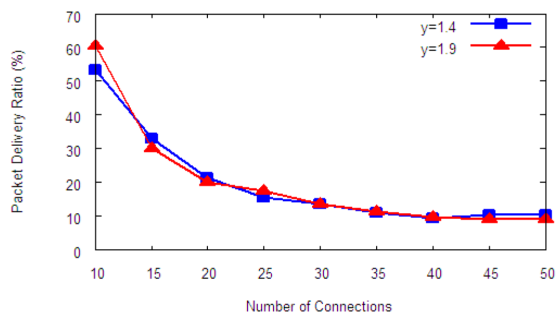


Figure 27: Throughput of New Algorithm (y=1.4 and 1.9)

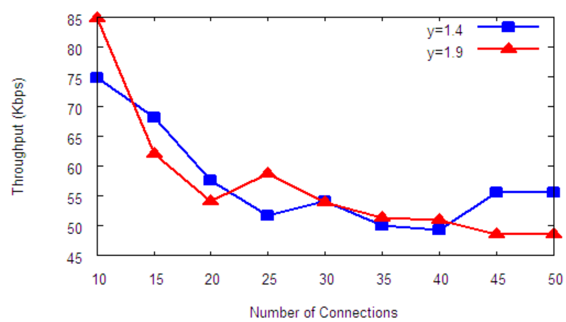
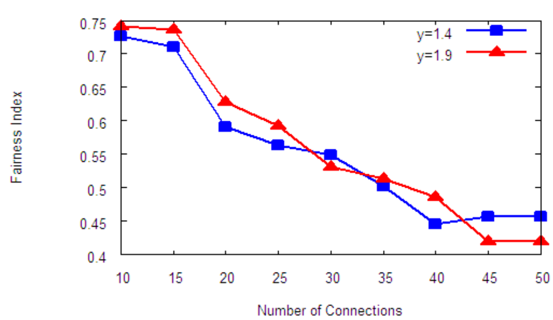


Figure 28: Fairness Index of New Algorithm (y=1.4 and 1.9)



6. Summary and Concluding Remarks

The backoff algorithm proposed in IEEE 802.11 in order to managing the contention for getting access to the channel causes many anomalies. IEEE 802.11 DCF and HBCWC scheme suffer from fairness problem in some configurations because nodes have the smallest CW have more chance to get access to the channel and when nodes have successful transmission, their priorities raises. This phenomenon causes fairness problem. HBCWC scheme also has significant performances such as end to end delay and throughput but it still suffers from fairness problem.

In this paper, we presented a new scheme that solves this fairness problem and eliminate fluctuations in HBCWC throughput. In addition, our new algorithm increases network overhead load significantly but it causes decrease in end to end delay. We get higher priorities to nodes had unsuccessful transmissions. This helps us to achieve fairness in the network.

Simulation results obtained using NS-2 shows the enhancement added by the new algorithm in term of fairness, network overhead load, and throughput. We use two parameters x and y for optimizing contention window. Simulation results also were presented for different value of y.

References

[1] S. Kumar, V.S. Raghavan and J. Deng, 2006. “Medium Access Control Protocols for Ad-Hoc Wireless Networks: A Survey”, Elsevier Ad-Hoc Networks Journal, Vol. 4, No. 3, pp. 326-358.

[2] J. Schiller, 2003. “Mobile Communications (2nd ed.)”, Addison Wesley.

[3] F. Cali, M. Conti and E. Gregori, 2000. “Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit”, IEEE/ACM Trans. on Networking, Vol. 8, No. 6, pp. 785-799.

[4] F. Cali, M. Conti and E. Gregori, 2000. “IEEE 802.11 Protocol: Design and Performance Evaluation of an Adaptive Backoff Mechanism”, IEEE Journal on Selected Areas in Communications, Vol. 18, No. 9, pp. 1774-1786.

[5] G. Bianchi, L. Fratta, and M. Oliveri, 1996. “Performance evaluation and enhancement of the CSMA/CA MAC protocol for IEEE 802.11 wireless LANs”, Proc. Seventh IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), vol. 2, pp. 392–396.

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

[7] G. Bianchi, I. Tinnirello, 2003. “Kalman filter estimation of the number of competing terminals in an IEEE 802.11 Network”, Proc. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), vol. 2, pp. 844-852.

[8] D.J. Deng, C.H. Ke, H.H. Chen and Y.M. Huang, 2008. “Contention Window Optimization for IEEE 802.11 DCF Access Control”, IEEE Trans. on Wireless Communications, Vol. 7, No. 12, pp. 5129-5135.

[9] L. Bononi, M. Conti, and E. Gregori, 2004. “Runtime optimization of IEEE802.11 wireless LANs performance”, IEEE Trans. on Parallel and Distributed Systems, vol. 15, no. 1, pp. 66–80.

[10] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, 1994. “MACAW: A Media Access Protocol for Wireless LAN’s”, Proc. ACM SIGCOMM Conference on Communications Architectures Protocols and Applications, pp. 212–225.

[11] M. Natkaniec and A.R. Pach, 2000. “An analysis of the backoff mechanism used in IEEE 802.11 networks”, Proc. fifth IEEE symposium on Computers and Communications, pp. 444–448.

[12] H. Wu, S. Cheng, Y. Peng, K. Long, and J. Ma, 2002. “IEEE 802.11 distributed coordination function (DCF): analysis and enhancement”, Proc. IEEE Int. Conference on Communications (ICC), vol. 1, pp. 605–609.

[13] P. Chatzimisios, et al, 2005. “A simple and effective backoff scheme for the IEEE 802.11 MAC protocol”, Proc. International Conference on Cybernetics and Information Technologies, Systems and Applications (CITSA), vol. 1, pp. 48-53.

[14] N. Song, B. Kwak, J. Song, and L.E. Miller, 2003. “Enhancement of IEEE 802.11 distributed coordination function with exponential increase exponential decrease backoff algorithm”, Proc. IEEE VTC2003 Spring, vol. 4, pp. 2775–2778.

[15] N. Song, B. Kwak, J. Song, and L.E. Miller, 2005. “Analysis of EIED backoff algorithm for the IEEE 802.11 DCF”, Proc. IEEE 62nd Vehicular Technology Conference (VTC), vol.4, pp. 2182—2186.

[16] Q. Ni, I. Aad, C. Barakat, and T. Turletti, 2003. “Modeling and Analysis of Slow CW Decrease for IEEE 802.11WLAN”, Proc. 14th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), vol. 2, pp. 1717-1721.

[17] I. Aad, Q. Ni, C. Castelluccia, and T. Turletti, 2002. “Enhancing IEEE 802.11 performance with slow CW decrease”, IEEE 802.11e working group document 802.11-02/674r0.

[18] A. Ksentini, A. Nafaa, A. Gueroui, M. Naimi, 2005. “Determinist contention window algorithm for IEEE 802.11”, Proc. IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), vol. 4, pp. 2712-2716.

[19] S. Romaszko, C. Blondia, 2006. “Dynamic distributed contention window control in wireless ad hoc LANs”, Proc. the Australian Telecommunication Networks and Applications Conference (ATNAC), vol. 18, no. 9.

[20] J. Deng, P.K. Varshney, and Z.J. Haas, 2004. “A new backoff algorithm for the IEEE 802.11 distributed coordination function”, Proc. Communication Networks and Distributed Systems Modeling and Simulation (CNDS), pp. 215-225.

[21] Y. Kwon, Y. Fang, H. Latchman, 2003. “A novel MAC protocol with fast collision resolution for wireless LANs”, Proc. IEEE the 22th IEEE International Conference on Computer Communications (INFOCOM), vol. 2, pp. 853–862.

[22] A. Balador, A. Movaghar, S. Jabbehdari, 2010. “History Based Contention Window Control (HBCWC) in IEEE 802.11 Mac Protocol in Error Prone Channel”, Journal of Computer Science (JCS), vol. 6, no. 2, pp. 205–209.

[23] NS-2 Simulator, http://www.isi.edu/nsnam/ns/, 2009.