



# **Minimizing Idle Time and Collision Time Using Token DCF With Append Process and Change Contention Window Size for Wireless Network**

Shailja Uikey, Namrata Sahayam

Department of Electronics Communication, Branch Communication System, Jabalpur Engineering College, Jabalpur,  
India.

**ABSTRACT:** IEEE 802.11 DCF is the MAC protocol currently used in wireless LANs. However, due to idle and collision times, 802.11 DCF performs poorly when it comes to channel utilization, system throughput, and channel access time. To overcome these sources of inefficiency in 802.11 DCF, we propose a distributed and dynamically adaptive MAC protocol for wireless networks, called Token-DCF with Append process and change contention window size. Main focus of our approach is on reducing idle and collision times by introducing an algorithm of token passing using Append process and change contention window size. In Append process strong data searched by DCF process and its queue length if greater than the threshold level, will focus on the minimum queue length in header list and add it with the new neighboring node by increasing system throughput and channel access time. We simulate in ns-2 to measure and compare performance of MAC protocols.

## **I. INTRODUCTION**

IEEE 802.11 defines the distributed coordination function (DCF) to share the wireless medium among multiple stations. DCF employs CSMA/CA with a binary exponential backoff algorithm to resolve channel contention. DCF specifies random backoff, which forces a station to defer its access to the channel for a random period of time. This backoff period corresponds to the number of idle slots a station has to wait before its transmission attempt. If multiple stations choose the same backoff, they will attempt to transmit at the same time and collisions will occur. Two types of overhead are associated with random access protocols. One is channel idle time (i.e., backoff time) which is the time when contending stations are waiting to transmit. Another is collision which happens when multiple stations transmit simultaneously. If there are few contending stations, idle time is the dominant overhead. If there are many contending stations, collision probability increases and becomes the main source of low channel utilization.

## **II. RELATED WORK**

We summarize the prior work into:

- 1) Token-DCF: An opportunistic MAC protocol for wireless networks Ghazale Hosseinabadi and Nitin Vaidya Department of ECE and Coordinated Science Lab. University of Illinois at Urbana-Champaign fghossei2, [nhvg@illinois.edu](mailto:nhvg@illinois.edu) 978-1-4673-5494-3/13/\$31.2013 IEEE
- 2) performance Analysis of Backoff Algorithm in IEEE 802.11 Networks Sakshi Suhane, Dr. Sanjeev Sharma, Prof. Varsha Sharma. IJSER © 2011 <http://www.ijser.org>. [2]
- 3) F. Cali, M. Conti, and E. Gregori, “Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit,” IEEE/ACM Trans. On Networking, vol. 8, no. 6, December 2000. [3].
- 4) Effect of Contention Window on the Performance of IEEE 802.11 WLANs Yunli Chen and harma P. Agrawal. Logarithmic Based Backoff Algorithm for MAC Protocol in MANETs Department of Computing Science University of Glasgow Glasgow G12 8RZ {Saher, [Mohamed](mailto:Mohamed@dcs.gla.ac.uk)}@dcs.gla.ac.uk.

The main design goal of Token-DCF is to reduce both idle time and collision time by introducing an implicit token passing algorithm. Token-DCF achieves 2X improvement in system throughput and channel access delay compared to 802.11 DCF for most network configurations [1].



# International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2014

The primary Medium Access Control (MAC) technique of IEEE 802.11 is called Distributed Coordination Function (DCF). This protocol adopts a Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) with a binary exponential backoff (BEB) algorithm to access the channel. The protocol performance mainly depends on backoff procedure which reduces the probability of collision. With BEB, waiting time of a node gets doubled after every unsuccessful transmission. This introduces fast-growing retransmission delays for the backlog traffic. DCF reduces the Contention Window to the initial value after each successful transmission which essentially assumes that each successful transmission is an indication that the system is under low traffic loading. In this paper analyzed the problem with existing Backoff Algorithm. [2]

Various MAC protocols have been proposed to improve the efficiency of DCF. Cali et al. modify the backoff algorithm of the IEEE 802.11 MAC protocol and derive a contention window size that maximizes network throughput [3].

Yunli Chen and Dharma P. Agrawal [4] a proposed in this paper, they use a fixed contention window (FCW) scheme and then evaluate the performance of this scheme. they analysis, determine an optimal contention window (OCW). Adlen Ksentini, Abdelhamid. Nafaa, Abdelhak Gueroui, Mohammed Naimi [5] a proposed, interval is dynamically controlled (increased/decreased) by the backoff algorithm. Setting the length of the backoff intervals is not a trivial task. Actually, when the network load increases, should increase the mean backoff interval (i.e., transmission differing time) to absorb the increasing number of contending flows, and hence minimizing the collision probability for those flows. Whereas, when the network load decreases, should decrease the mean backoff interval, which reduces the spacing between successive frame transmissions; large values of backoff may indeed strongly limit the throughput of fewer backlogged flows.

### III. TOKEN-DCF DESIGN

In Token-DCF, when a station transmits on the channel, it might give a privilege (i.e., a token) to one of its neighbors. When a transmission ends, the privileged station, starts transmitting after a short period of time, namely SIFS (Short Inter Frame Space). Non-privileged stations follow the backoff procedure of 802.11 to access the channel. Token-DCF is implemented in the MAC layer of the protocol stack. Scheduling information is embedded in the MAC header of data packets and is transferred to the neighboring stations via overhearing. Each station maintains queue length of the neighboring stations. These queue lengths are then used in the scheduling phase to select the privileged station for the next transmission. Transmitting station announces the privileged station in the privileged field of the MAC header of the data packets it transmits. By overhearing these packets, the privileged station is informed that it has a higher priority for the next transmission. When a transmission ends, the privileged station can start transmitting after SIFS if the channel is sensed idle. If opportunistic overhearing does not work, i.e., token is not received by the next privileged station, Token-DCF operates similar to 802.11 DCF. But when the next privileged station overhears the token, it can transmit on the channel without going to the backoff procedure.

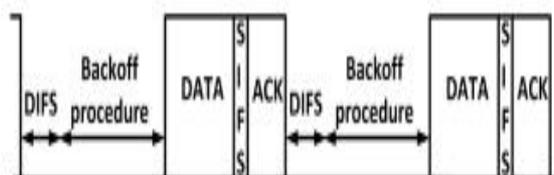


Fig. 1: Access method of IEEE 802.11 DCF



# International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2014

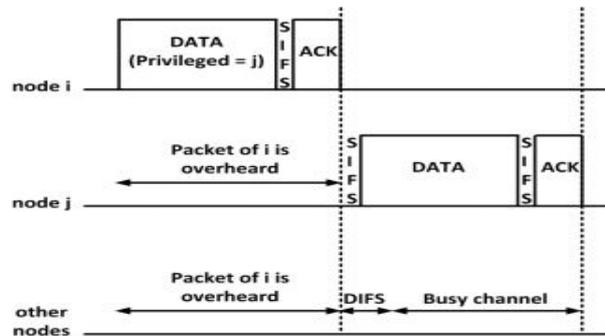


Fig. 2: Access method of Token-DCF protocol

## IV. APPEND PROCESS

Next neighboring node is searched [strong data] by DCF process. If its queue length is greater than the threshold level, collision will occur and the data in the node will be lost. We will focus on the minimum queue length in header list and add it with the new neighboring node and add their information like node id and queue length in the header list. Then again privileged process will start, and continuity of the process will remain same.

## V. CONTENTION WINDOW

In every station there is a contention window, which is storing and forwards the temporary data. Contention window come up with two categories, contention window minimum and contention window maximum, and its range in between 32 to 1024, which is distributed uniformly in every station. [2]

When the number of node packets going to be transferred, so by the backoff procedure all nodes select backoff slots and it is going to decrease to 32 minimum and when it is reached to minimum slots, it is transmitted the data accordingly.

When packets have been successfully transferred so, it is set the 32 minimum range as a range of contention window. When the packets are not properly transferred, it makes size of contention window doubled. [2] Like 32, 64, 128, 256, 512, and 1024 this is called binary exponential backoff (BEB) algorithm. Successful transition =  $CW_0 = CW_{min} = 32$  Unsuccessful transition =  $CW \times 2 = CW_{32 \times 2}$ .

### Backoff stage –standard range formula-

$$B_i = 2^{(I+k)} - 1$$

$B_i$  = Basic DCF

$B_i \times$  slot time

$I$  = initially equal to 1

$K$  = depend on the physical layer parameter.

$i=1$

$$B(1) = 2^{(1+4)} - 1$$

$$B(1) = 2^{(5)} - 1$$

$$B(1) = 2 \times 2 \times 2 \times 2 \times 2 - 1$$

$$= 32 - 1$$

$$= 31$$



# International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2014

## VI. EVALUATION

In the example considered, the threshold level of each node is assumed to be 200 bytes, i.e., each node can contain 200 bytes of data. Suppose we run DCF process for 10 nodes.

Example-

20, 30, 40, 50, 160, 170, 180, 185, 190 and 200.

Then checks the total receives size of data and calculates the throughput.

$=20+30+40+50+160+170+180+185+190+200 = 1225$

**[THROUGHPUT = (NUMBER OF PACKETS RECEIVE / NUMBER OF PACKETS GENERATED X 8)  
BPS]**

10 node = generated bytes (2000)

Throughput =  $1225/2000 = (0.6125 \times 8) = 4.9$  bps.

Then Source node searches for the node with the largest queue length, i.e., the node with largest data, in its header list.

Once found, the node with largest queue length is granted privilege.

Largest Queue length = 200

Now privilege (token) process is run for this node. The data contained inside the node is distributed among each selected node.

20, 30, 40, 50, 160

The average is calculated by dividing the total bytes of data, inside that node, by total number of selected node.

The selected node, among which the data is distributed, is the nodes which have minimum bytes of data. Reason for doing this is that if data is add to nodes which already have enough data, it may exceed the threshold level, and thus collision of node may occur. To prevent this, we select a predefined number of nodes, in our example nodes are selected by sorting out of 10 nodes on the basis of byte of data they contain, the first 5 nodes, having least data, are taken.

**[Average = Total bytes of data/ Total no. of selected node]**

Average =  $200/5 = 40$

$20+40=60, 30+40=70, 40+40=80, 50+40=90, 160+40=200,$

$170, 180, 185, 190, 150$

Then again we check the total receives size of data and calculate the throughput.

Receives size =  $60+70+80+90+200+170+180+185+190+150 = 1375$

Throughput =  $1375/2000 = (0.6875 \times 8) = 5.5$  bps.

The performance is better than the simple DCF process. So, privileged process is improving the efficiency of simple DCF process.

After running the privilege process for limited number of time, the result is checked and throughput is calculated.

In special cases if a new neighboring node is encountered which has queue length greater than threshold level, then append process is used. In this process the bytes of data is the new neighboring node is checked. If the bytes are above the threshold level then the data equal to the threshold level is added to the new neighboring node and the remaining part of bytes of data is add with the data of the node with minimum queue length. After running the append process for the limited number of time, the result is checked and throughput is calculated. After completing the append process, then again privilege process is repeated.

We run DCF process for 10 nodes.

20, 40, 60, 80, 50, 120, 140, 160, 180 and 250.

In append process, if a new neighboring node is searched with strong data whose bytes is above 200 bytes, for instance 250 bytes, then the new neighboring node stores the 200 byte. The node with minimum byte of node is searched in header list and the remaining 50 bytes of data is added to it.

$20+50=70, 40, 60, 80, 50, 120, 140, 180, 200.$

Then again privilege process is repeated.



# International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2014

We know that, for less number of nodes, the performance decreases due to idle time, because the size of the contention window is more compared to the nodes. The minimum node chooses the slots of longer distance and fails to decrease at a given time due to which transmission fails. Nodes are backoff for retransmission, due to this the minimum size of the contention window is doubled and the delay increases.

At the start the size of the contention window is taken 32. Because of this, the performance of node number 5,10,15,20 decreases. To increase the performance and decrease the delay the size of the contention window is decreased from 32 to 16 with Append process. Now the minimum nodes easily choose backoff slots and decrease and successful transmission occurs in the given time.

Backoff stage –standard range formula-

$$B_i = 2^{(I+k)} - 1$$

i=0

$$B(1) = 2^{(0+4)} - 1$$

$$B(1) = 2^4 - 1$$

$$B(1) = 2 \times 2 \times 2 \times 2 - 1$$

$$= 16 - 1$$

$$= 15$$

Figure 5 shows the performance of the three protocols. Performance is defined as the modified Token DCF (Append process with change contention window size) as compared to Token DCF and IEEE 802.11 Mac protocol.

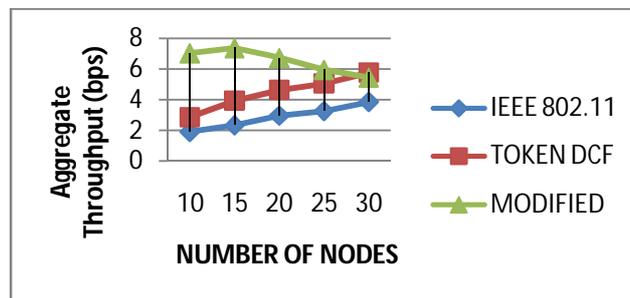


Fig. 3: Aggregate Throughput (area = 150m x 150m)  
Contention window minimum size= 16

The performance of nodes number 50, 55 and 60 decreases. To increase the performance the size of the contention window is increased from 16 to 64 with Append process. The maximum nodes easily choose backoff slots and decrease and successful transmission occurs in the given time.

Also, there remains no need to use the Binary Exponential backoff algorithm and the delay does not increase.

Backoff stage –standard range formula-

$$B_i = 2^{(I+k)} - 1$$

i=2

$$B(2) = 2^{(2+4)} - 1$$

$$B(2) = 2^6 - 1$$

$$B(2) = 2 \times 2 \times 2 \times 2 \times 2 - 1$$

$$= 64 - 1$$

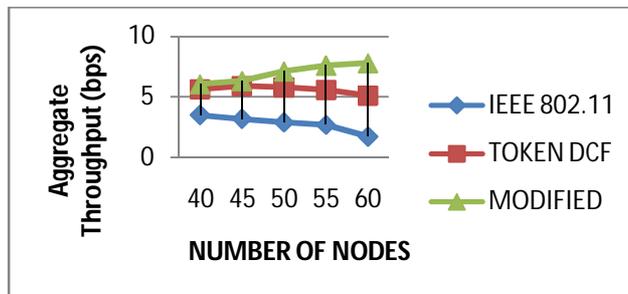
$$= 63$$

Figure 4 shows the performance of the three protocols. Performance is defined as the modified Token DCF (Append process with change contention window size) as compared to Token DCF and IEEE 802.11 Mac protocol.

# International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

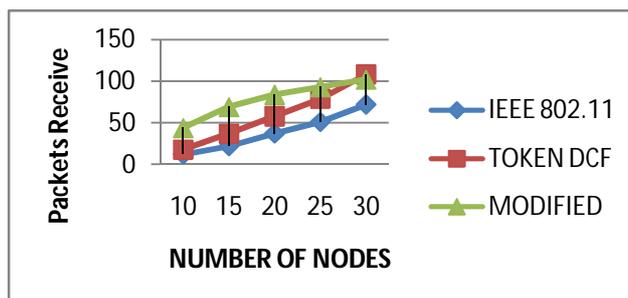
(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2014



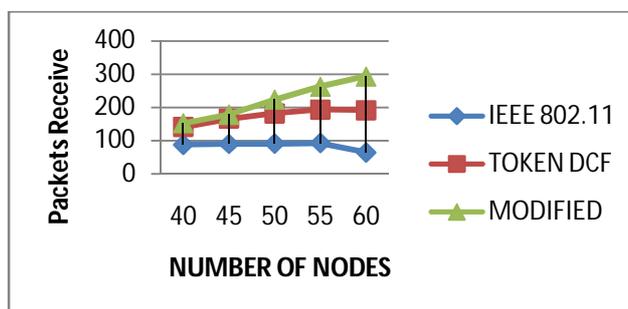
**Fig. 4: Aggregate Throughput (area = 150m x 150m)  
 Modified (Append process) with Contention window min size= 64**

Figure 5 shows the packets receive of the three protocols. Packets receive is defined as the number of packets receive in modified Token DCF (Append process with change contention window size) as compared to Token DCF and IEEE 802.11 Mac protocol.



**Fig. 5: Packets receive (area = 150m x 150m)  
 Modified (Append process) with Contention window min size= 16**

Figure 6 shows the packets receive of the three protocols. Packets receive is defined as the number of packets receive in modified Token DCF (Append process with change contention window size) as compared to Token DCF and IEEE 802.11 Mac protocol.



**Fig. 6: Packets receive (area = 150m x 150m)  
 Modified (Append process) with Contention window min size= 64**

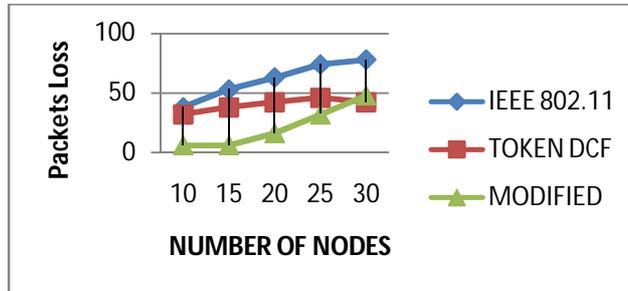
Figure 7 shows the packets loss of the three protocols. Packets loss is defined as the number of packets loss in modified Token DCF (Append process with change contention window size) as compared to Token DCF and IEEE 802.11 Mac protocol.



# International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

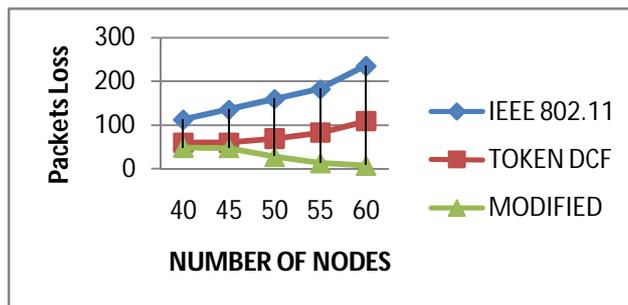
(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2014



**Fig. 7: Packets loss (area = 150m x 150m)  
Modified (Append process) with Contention window min size= 16**

Figure 8 shows the packets loss of the three protocols. Packets loss is defined as the number of packets loss in modified Token DCF (Append process with change contention window size) as compared to Token DCF and IEEE 802.11 Mac protocol.



**Fig. 8: Packets loss (area = 150m x 150m)  
Modified (Append process) with Contention window min size= 64**

## VII. CONCLUSION

In this paper, we presented the design and performance evaluation of Token-DCF using Append process with change contention window size. Token-DCF using Append process is a distributed MAC protocol that uses an opportunistic overhearing mechanism to schedule network stations for transmission on the channel and we changed the size of contention window according to the node means increasing and decreasing the size of contention window. Thus we concluded that modified Token DCF, improves efficiency, reduces idle time, collision time, packet loss and increases throughput.

## REFERENCES

- [1] Token DCF: An opportunistic MAC protocol for wireless networks. 2013 IEEE.
- [2] performance Analysis of Backoff Algorithm in IEEE 802.11 Networks Sakshi Suhane, Dr. Sanjeev Sharma, Prof. Varsha Sharma. IJSER © 2011 <http://www.ijser.org>.
- [3] F. Cali, M. Conti, and E. Gregori, "Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit," IEEE/ACM Trans. On Networking, vol. 8, no. 6, December 2000.
- [4] Effect of Contention Window on the Performance of IEEE 802.11 WLANs Yunli Chen and harma P. Agrawal. Logarithmic Based Backoff Algorithm for MAC Protocol in MANETs Department of Computing Science University of Glasgow Glasgow G12 8RZ {Saher, [Mohamed](mailto:Mohamed@dcs.gla.ac.uk)}@dcs.gla.ac.uk.
- [5] Determinist Contention Window Algorithm for IEEE 802.11. Adlen Ksentini, Abdelhamid. Nafaa, Abdelhak Gueroui, Mohammed Naimi.
- [6] Performance Analysis of the 802.11 Distributed Coordination Function under Sporadic Traffic M. Garetto and C.-F. Chiasserini Dipartimento di Elettronica, Politecnico di Torino, Italy Email: [fgaretto](mailto:fgaretto@polito.it), [chiasserini@polito.it](mailto:chiasserini@polito.it).
- [7] M. Ergen, D. Lee, A. Puri, P. Varaiya, R. Attias, R. Sengupta, S. Tripakis, "Wireless token ring protocol," Proceedings of the Eighth IEEE Symposium on Computers and Communications (ISCC 2003) pp. 710-715, 2003.
- [8] A. Dimakis and J. Walrand. "Sufficient conditions for stability of longest-queue-first scheduling: Second-order properties using fluid lim-its." Advances in Applied Probability, 38(2):505521, 2006.
- [9] S. Shakkottai, R. Srikant, and A. Stolyar. "Pathwise optimality of the exponential scheduling rule for wireless channels," Advances in Applied Probability, 36:10211045, 2004.
- [10] A. Eryilmaz, R. Srikant, and J. Perkins. "Stable scheduling policies for fading wireless channels," In Proceedings of IEEE International Symposium on Information Theory, 2003