Improving the Performance of the IEEE 802.11 Distributed Coordination Function

Asad Ali, *Member IACSIT*

***Abstract*—IEEE 802.11 is the most popular Wireless LAN (WLAN) system in the world today. The primary medium access control (MAC) technique of 802.11 is distributed coordination function (DCF). DCF is a carrier sense multiple access with collision avoidance (CSMA/CA) scheme with binary slotted exponential back-off. In DCF of IEEE 802.11, The Request to send/Clear to send mechanism is very effective in terms of system performance, but this mechanism may lead to underutilization of resources as well, when either RTS or CTS fails to reach their respective destinations. In such a case, the nearby nodes of both the sender and receiver, after Updating their Network Allocation Vector (NAV) remain idle for the whole NAV period, even if there is no transmission on the channel, thus leading to a major wastage of the resources. So in this paper we propose a mechanism to overcome this problem, so that the time may be saved and the resources may be utilized efficiently.**

***Index Terms*—Network allocation vector (NAV), RTS/CTS, distributed coordinated function (DCF).**

I. INTRODUCTION

Wireless communication has also greatly influenced our lives. Wireless technology is rapidly becoming a crucial component of computer networks and its use is growing by leaps and bounds. Today IEEE 802.11 is the most popular Wireless LAN (WLAN) system in the world. In IEEE 802.11 Distributed Coordination Function (DCF) is the fundamental mechanism to access the medium [1][2]. DCF scheme is based on the Carrier senses multiple accesses with collision avoidance (CSMA/CA). The collided packets are retransmitted according to the binary exponential backoff rules. DCF describes two techniques for packet transmission. Two way handshaking also called basic access mechanism, in which the source node sends a data packet and the destination responses by immediate transmission of the acknowledgement (ACK), if the data is successfully received by the destination node. The other access mechanism is four way handshaking also known as request to send/clear to send (RTS/CTS). In RTS/CTS, when a node wants to send a data packet, it does so by first reserving the channel for data transmission by sending RTS to the destination [1], [2], [5], [6]. The destination station acknowledges the receipt of an RTS frame by sending back a CTS frame, after which normal packet transmission and ACK response occurs. RTS/CTS include the expected time for which the two nodes will be exchanging data and thus the time for which the channel will be busy. The neighboring nodes of sender and receiver, overhearing the RTS and CTS defer their transmission for the time specified in the RTS/CTS. For this reason, each host maintains a variable called the Network Allocation Vector (NAV) which is updated on the basis of the RTS/CTS. This actually mean that the neighboring nodes, for the time specified in the NAV which is updated on the basis of the RTS/CTS, will go idle and will not sense the channel for the whole NAV period. Therefore, when a station is hidden from either the transmitting or the receiving station, by detecting just one frame among the RTS and CTS frames, it can suitably delay further transmission, and thus avoid collision.

Rest of the paper is organized as follows. Section II contains the related work, Section III contains the problem statement, Section IV highlights our proposed scheme, Section V shows the result and Section VI shows the conclusions.

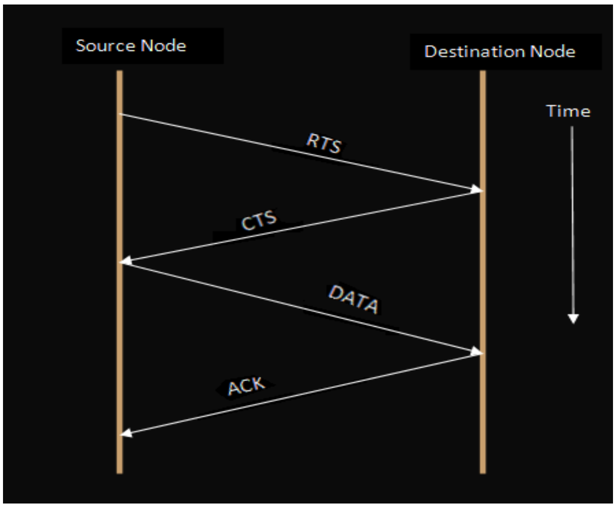

Fig. 1. Four way handshaking access mechanism.

II. PROBLEM STATEMENT

The neighboring nodes of both the sender and the receiver, upon receiving the RTS/CTS, update their NAV and thus defer their transmission for the time specified in the RTS/CTS. As mentioned earlier, for data communication sender sends RTS to the destination to inform him of the subsequent data communication. Thus upon receiving this RTS the neighbor of the sender update their NAV and defer their transmission and thus go idle. Similarly, when destination wants to acknowledge by sending CTS, the neighbor of the destination node, upon receiving this CTS, update their NAV. So they too, will go idle and will not sense the channel. Thus a collision free transmission takes place. However, if the RTS fails to reach the destination or the CTS fails to reach the source, in that case there will be no data

Manuscript received August 10, 2012; revised October 3, 2012.

Asad Ali is with the School of Electrical Engineering and Computer Science, National University of Science and Technology (NUST), Islamabad, Pakistan (e-mail: asad.ali@seecs.edu.pk).

transmission. Yet the neighboring nodes of both the sender and the receiver, who have updated their NAV, will still remain idle and will not sense the channel. Though there is no transmission in actual sense but all the neighboring nodes are still idle and thus leading to a major underutilization of resources and thus overall degrading the performance of the 802.11 network. One reason of no CTS from destination is that destination node cannot send CTS due to Information Asymmetry (IA), defined by Garreto in [3].

Consider the Fig. 2 below, if A sends RTS which is only received by the neighbors of A or as mentioned above, it is received by B as well, but due to some reason like the one discussed above (IA), B cannot send CTS in reply. So in that case the neighbors of the A will remain idle for the whole backoff period after updating the NAV. Similarly if B sends CTS and it fails to reach A and is only received by B's neighbor than in that case the neighbors of B will also go idle after updating their NAV upon receiving CTS.

Fig. 2. Data transmission from A to B.

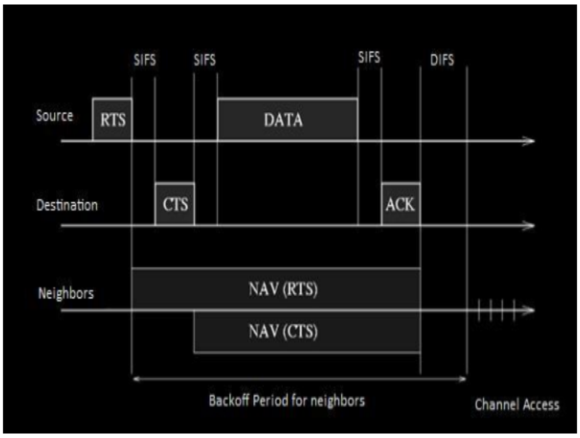

Fig. 3. RTS/CTS access mechanism.

III. RELATED WORK

With the popularity of the IEEE 802.11, lot of research work has been done to analyze the throughput of 802.11.[1] has presented an analytical model compute the saturation throughput performance of the 802.11 Distributed Coordination Function (DCF) with the assumptions of finite number of terminals and ideal channel conditions. The proposed analysis applies to both the packet transmission schemes employed by DCF, namely, the basic access and the RTS/CTS access mechanisms. He has given the transmission probability as

$$P_{tr} = 1 - (1 - \tau)^n \tag{1}$$

He has given the probability of successful transmission as

$$P_s = \frac{n\tau(1-\tau)^{n-1}}{P_{tr}} = \frac{n\tau(1-\tau)^{n-1}}{1-(1-\tau)^n} \tag{2}$$

The model and matrix in [5] provide a clear understanding of the behavior of CSMA protocols in arbitrary topologies as well as support the design of effective protocol solutions to the starvation problem. A novel analytical model that is able to compute the individual throughputs of all the flow in an arbitrary network topology is worked out. In [2] the author, has calculated the throughput of the 802.11 under the saturated traffic condition, and consider the effects of packet size, the number of contention nodes, transmission collision Probability and channel condition. Different authors have taken different assumption and different constraints to calculate the throughput of the 802.11 networks and the throughput of the DCF. But no one has calculated the time that could be saved if nodes do not go idle for the whole backoff period, in case of RTS or CTS failure and thus increase the throughput by utilizing the time saved by resetting backoff timers by neighboring nodes.

IV. PROPOSED SOLUTION

According to our proposed scheme, upon receiving RTS or CTS, the neighboring nodes of both the sender and receiver will not go permanently idle for the whole backoff period, after updating the NAV. In fact they will go idle only temporarily. Once the found that there is a transmission between the two nodes, only then, they will go permanently idle for the whole backoff period. Again referring two the Fig. 2 above, if RTS fails or there is no CTS from the destination, then the neighboring nodes of A will remain idle for the whole backoff period. If CTS fails to reach the source and is only received by the neighboring nodes of the destination i-e B, the neighbors of B will go idle for the whole backoff period, Thus a clear underutilization of resources. So in order to overcome this problem, our proposed scheme is as follows.

A. If RTS fails or No CTS

If RTS fails to reach the B or B is unable to send CTS, the neighbors will go idle temporarily for the time equal to 2SIFS+CTS+δ, as is clear from the Fig. 4 below. Here δ is the propagation delay. At this point, neighbors will sense the channel whether it is busy or not. If they find transmission on the channel, they will go permanently idle for the whole backoff period. Otherwise, they reset their backoff period.

$$\text{Total Idle time for A's Neighbors} = \text{SIFS}+\text{CTS}+\delta + \text{SIFS} + \text{Header(MAC+PHY)}+\text{DATA}+\delta + \text{SIFS} + \text{ACK} + \delta \tag{3}$$

$$\text{Time for temporary idle state} = 2\text{SIFS}+\text{CTS}+\delta \tag{4}$$

Time saved, if not transmission after temporary idle state

$$\text{Time saved} = \text{Header (MAC+PHY)} + \text{DATA} + \delta + \text{SIFS} + \text{ACK} + \delta \tag{5}$$

Equation (5) shows how considerable amount of time is saved, if there is no transmission due to RTS failure or due to lack of response on behalf of destination.

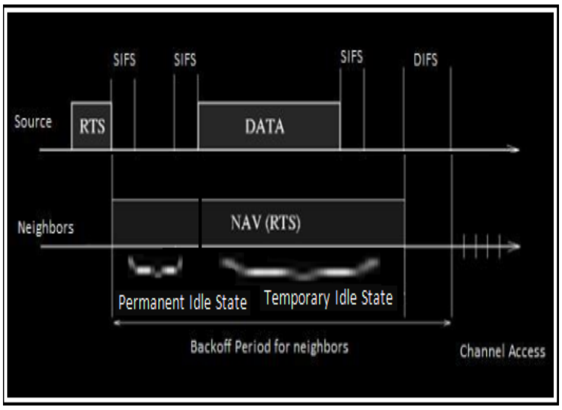

Fig. 4. Backoff period for sender's neighbors.

B. *If CTS Fails*

If due to some reasons, CTS fails to reach the destination and is only received by the neighbors of the destination, in that case neighbors will go temporarily idle for the time equal to SIFS+δ after receiving the CTS. At this point of time, the neighbors of B i-e destination, will sense the channel, if it is busy or not. If they found it to be busy, they will go permanently idle for the whole backoff period otherwise they will reset their backoff timers according to their transmission.

Total idle time for B's neighbors= SIFS + Header (MAC+PHY)+DATA+δ+SIFS+ACK+δ (6)

Temporary idle time =SIFS+δ (7)

Time saved, if not transmission after temporary idle state
Time Saved= Header (MAC+ PHY) + DATA + δ + SIFS + ACK (8)

Equation (8), shows the time saved if CTS fail to reach the destination and after the temporary idle time, if there is not transmission on the channel.

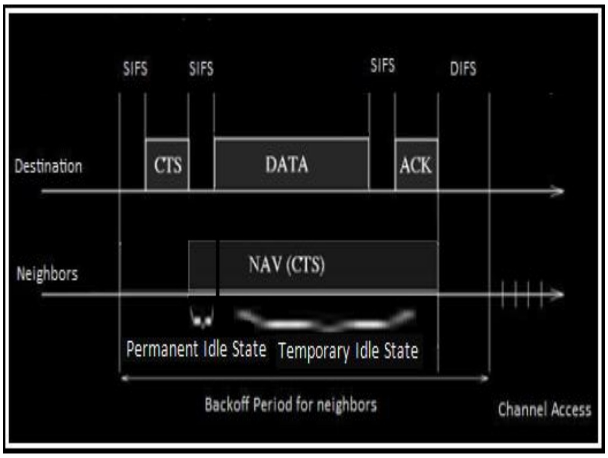

Fig. 5. Backoff period for destination's neighbors.

V. RESULTS

In this section, we will show the efficiency improved due to our proposed scheme. We will show that as the packet size gets larger, the time saved due to RTS/CTS lost also gets larger. And also as the error rate increases, the efficiency also improves accordingly. The parameters used for the analysis are shown in the table below. We have taken the same parameters as in [2].

TABLE I: SYSTEM PARAMETERS USED FOR ANALYSIS

SIFS	*10 μ s*
CTS	*304 bits*
ACK	*304 bits*
Propagation delay	*2 μ s*
Channel bit rate	*1Mbps*
PHY header	*24 bytes*
MAC header	*28 bytes*

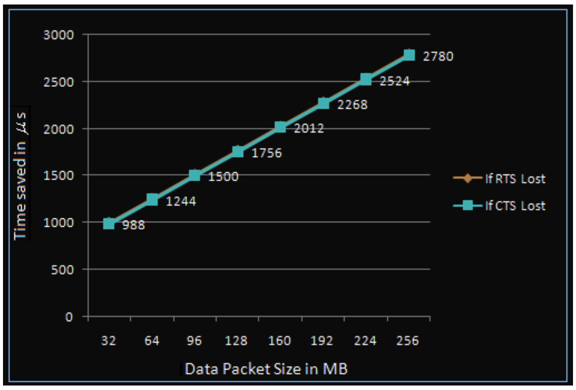

Fig. 6. Size of the data Vs time saved in μ s, if RTS or CTS is lost and permanent idle state is avoided

Fig. 6 above shows that as we increase the size of the data packet, the corresponding idle time will be decreased and thus the time saved increases highly as we increase the size of the data packet. The saved time for RTS and CTS is almost same. The curve for the time saved if CTS is lost overlaps the curve for time saved due to RTS loss and therefore the curve for RTS loss almost invisible in the Fig. The saved time is directly proportional to the data packet size. The time saved by avoiding the neighboring nodes go idle for the whole backoff periods reaches up to 98.18% if the data size become 2MB, if we make them go idle only temporarily for the time equal to SIFS+CTS+μ. So this significant amount of time can be used by other nodes to transmit their data, instead of remaining idle.

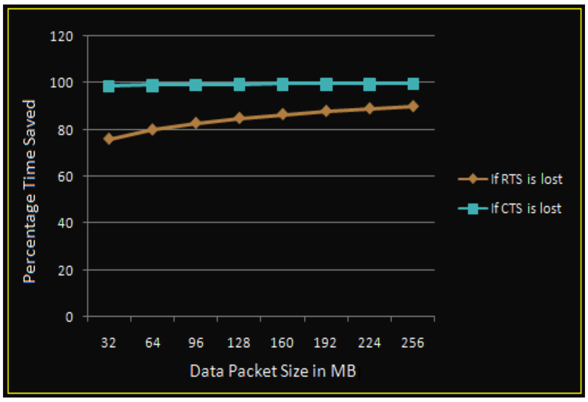

Fig. 7. Size of the data Vs percentage time saved, If RTS or CTS is lost and permanent idle state is avoided

Fig. 7 above shows that as we increase the data packet size, the corresponding percentage saved time for RTS and CTS loss also increases. The percentage time saved for CTS loss is slightly higher then the RTS loss. The time saved due to CTS loss almost reaches 100% for the data above then 2MB. At the data length of 2MB, the saved time is 99.92%. Thus we can save up to 100% time if we avoid idle state for the whole backoff period and make the neighboring nodes go idle only temporarily for the time equal to SIFS+ μ .

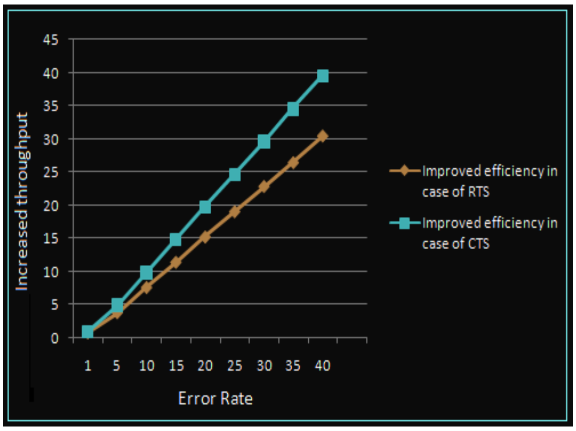

Fig. 8. Error rate Vs increased throughput, if RTS or CTS is lost and permanent Idle condition is avoided

Fig. 8 above shows that as the error rate increases, since the saved time also increases thus the overall throughput of the network will also increase considerably. In case if CTS is lost, the idle time for the neighboring nodes of destination node is lesser then the neighboring nodes of source, thus their throughput is greater then that of the source node's neighbors.

VI. CONCLUSION

We have shown a fine technique that considerably reduces the backoff period, if either RTS or CTS is lost. We have shown a technique that improves the performance of the IEEE 802.11 DCF by utilizing the time which was initially wasted when the nodes go idle for the whole backoff period even if either RTS or CTS is lost.

ACKNOWLEDGEMENT

I'm thankful to Dr.Anjum Naveed for his assistance in writing this paper. He has always welcomed us whenever we have sought his help. He removed lot of our mistakes and gave us number of good ideas to improve our research.

REFERENCE

[1] G. Bianchi "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, March 2000.

[2] J. Yin, X. D. Wang, and D. P. Agrawal, "Optimal packet size in error-prone channel for IEEE 802.11 distributed coordination function," *WCNC 2004 / IEEE Communications Society*

[3] M. Garetto, T. Salonidis, and E. W. Knightly, "Modeling per-flow throughput and capturing starvation in CSMA multi-hop wireless networks," *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*.

[4] M. Garetto, J. Shi, and E. W. Knightly, "Modeling media access in embedded twoflow topologies of multihop wireless networks," MobiCom'05, August 28–September 2, 2005, Cologne, Germany.

[5] M. Garetto, T. Salonidis, and E. W. Knightly, "Modeling per-flow throughput and capturing starvation in CSMA multi-hop wireless networks," *Infocom 2006. 25th IEEE International Conference on Computer Communications. Proceedings*.

[6] X. Chen, D. Qiao, J. Yu, and S. Choi, "Probabilistic-based rate adaptation for ieee 802.11 wlans," *IEEE Globecom 2007 Proceedings*