

Ad Hoc Operations of Enhanced IEEE 802.11 with Multiuser Dynamic OFDMA under Saturation Load

Hasan S. Ferdous and Manzur Murshed, *Member, IEEE*

Gippsland School of Information Technology, Monash University, Churchill, VIC 3842, Australia

Email: {Ferdous.Hasan, Manzur.Murshed}@infotech.monash.edu.au

Abstract—In this paper, we discuss the challenges associated with integrating multiuser OFDMA in a single cell IEEE 802.11 based wireless ad hoc network and propose a new, dynamic and robust approach to improve it. Our new MAC, using OFDMA in the physical layer, can incorporate multiple concurrent transmissions or receptions in a dynamic manner and can adjust the collision probability based on the traffic load when nodes are endowed with a single half-duplex radio only. Simulation results show that for moderate number of users, our system improves throughput by up to 20%, decreases collision in control messages by up to 45% and reduces the average delay by up to 18%.

I. INTRODUCTION

IEEE 802.11 based wireless local area networks (WLANs) have been ubiquitous nowadays to provide last hop wireless connectivity at homes, offices, and public places. Although access point (AP) based WLANs are very popular and widely used, there is another mode of operation that is gaining popularity among the users. Ad hoc networking allows the users to connect to the network seamlessly, without the need of setting and configuring an access point. This type of networks are particularly useful in a scenario where we need to set up and use a network on the go - like in a battle field or in a situation combating a disaster. In ad hoc networks, there is no central coordinator; nodes collaborate among themselves to facilitate the communication. Any node can send data to any other node using IEEE 802.11 distributed coordination function (DCF) as the medium access control (MAC) layer protocol. As new applications of these networks are increasing steadily, the issue of improving their performance is becoming more and more important.

In his seminal paper [1], Bianchi proposed a very accurate analysis of the IEEE 802.11 DCF using a discrete-time Markov chain model. Based on the assumption of constant and independent collision probability of a packet transmitted by each station, Bianchi found the probability τ that a station transmits in a randomly selected slot and the collision probability p , given a station transmits. It follows from his analysis that IEEE 802.11 DCF is not very efficient in terms of utilizing the available bandwidth, especially when the number of contending nodes are high.

In [2], Xiao *et al.* analyzed the performance of IEEE 802.11 DCF and formulated a mathematical limit for the achievable throughput and delay of it. They proved that there is a throughput upper limit and delay lower limit that depend on the data packet length and are independent of the data transmission rate. Their formula matches almost perfectly with

the practical achievable throughput of IEEE 802.11 DCF, which is almost 50% of the theoretical capacity of the network.

We can understand the reasons behind this inefficiency of IEEE 802.11 DCF. Wireless channel is inherently broadcast in nature. Nodes use half-duplex radios; so they need RTS/CTS mechanism to avoid collisions for large data packets. Again, physical layer convergence protocol (PLCP) header and preamble as well as all the control messages like RTS, CTS, and ACK are sent at the lowest data rate (6 Mbps) so that all the nodes can decode these messages while data can be sent at a maximum rate of 54 Mbps (802.11a/g). So, these control messages, though quite small, occupy a significant portion of the transmission time available. According to the analysis shown in [1], RTS collision probability is not negligible, even for moderate number of nodes. Time is also wasted during backoff, DIFS, and SIFS intervals. For every successful packet transmission, 195 μ s is wasted to exchange these control messages and inter frame spacing times, in addition to the backoff period. Collisions also increase the wasted time in backoff by increasing the contention window size. For every RTS collision, at least another 98 μ s is lost. This wastage is not negligible when the number of nodes increases and we need to find a way to reduce this inefficiency of IEEE 802.11 DCF.

Recent research works on the capacity of wireless networks [3], [4] showed that incorporating concurrency at the physical layer is one of the best ways to make wireless networks scalable. Multiple input multiple output (MIMO) can do this by enabling the devices to receive n data streams in parallel or eliminate interference from as many as $(n-1)$ neighbors. But it requires the nodes to have at least n antennas and consumes higher energy. Another approach to obtain concurrency is to use OFDMA (Orthogonal Frequency Division Multiple Access). In OFDMA systems, the frequency band is divided into some non-overlapping sub-channels consisting of orthogonal subcarriers. It can enhance the efficiency in the physical (PHY) layer by increasing multiuser diversity and decreasing peak to average power ratio. Using OFDMA, we can incorporate multiple concurrent transmissions or receptions in IEEE 802.11 DCF and improve the performance of it to a significant extent.

There are many works on synchronization [5], spectrum allocation [6] and signaling [7] in OFDMA based systems. But few works discussed about implementing OFDMA in IEEE 802.11. Valentin *et al.* [8] discussed the issues that are needed to be considered to implement OFDMA in IEEE 802.11. An-

other paper of them [9] described a practical implementation of their proposed system in a real software radio test-bed and showed the effectiveness of their design. They discussed about the MAC and LLC (Logical Link Control) sublayer extensions required for implementing OFDMA, a fair and optimized policy for physical protocol data unit (PPDU) fragmentation, and sub-channel allocation for the nodes. The main drawback of their work is that they considered only downlink traffic, i.e., data can be sent from the AP to the wireless stations only.

Veyseh *et al.* [10] proposed their DCF named CTRMA (Concurrent Transmission or Reception Multiple Access) and compared their work with single-radio single-channel, single-radio multi-channel, and multi-radio multi-channel MAC protocols and found the superiority of their proposal in terms of network throughput. The main drawback of their CTRMA protocol is that, it requires extensive message exchange between the users, in order of $O(d^2C)$ where d is the average number of neighbors and C is the number of sub-channels. Again, message exchange is required every time the network topology changes. They showed very little experimental results and their performance comparison is based on network throughput only. They did not discuss about the average delay or other performance metrics of their system. They assumed a separate control channel for RTS/CTS exchange and information negotiation, which decreases the available data bandwidth. This control channel is narrower than the data channel. So, probability of congestion and collisions in this channel may increase. In another paper, Veyseh *et al.* [11] proposed another DCF based on OFDMA that can incorporate multiple concurrent transmissions or receptions. But this proposal is based on a receiver initiated collision avoidance protocol, which is not used in traditional wireless devices.

In our earlier work [12], we showed some insight about the collision avoidance scheme of IEEE 802.11 DCF and pointed the inherent reasons for its inefficiency. With the help of *Birthday Problem*, we proved that sub-channelization can decrease collisions in RTS messages to a great extent. We proposed an OFDMA based MAC for AP based operations of IEEE 802.11 that can support both uplink and downlink. We empirically showed the superiority of our proposed DCF in terms of throughput and delay for saturation load. In this paper we will extend our proposed DCF for ad hoc operations of IEEE 802.11 based WLANs.

In our proposed DCF, we divide n nodes into c groups and assign a sub-channel for each of them such that the i -th group has $n_i = \lceil n/c \rceil$ nodes, for $i \leq n \bmod c$, or $\lfloor n/c \rfloor$ nodes, otherwise. Nodes in each group transmit RTS message using their assigned sub-channel. The contention cycle ends after each group has transmitted an RTS or a specified timeout occurs. The recipients respond with CTS using the sub-channel at which they have received that RTS message. Then the nodes divide the channel into equal sub-channels based on the number of CTS messages and transmit data in parallel. We have shown that by reducing collisions and saving inter frame spacing times, our proposed DCF can improve the

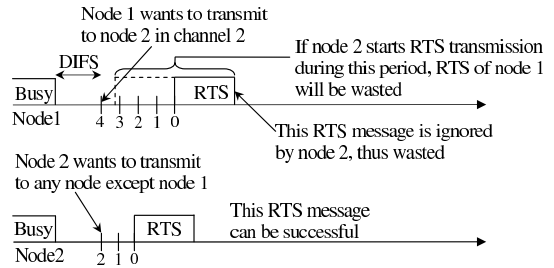


Fig. 1: Receiver end virtual collisions.

performance of the standard IEEE 802.11 DCF, specially when the number of contending devices is high.

Integrating OFDMA for ad hoc operations poses additional problems and concerns that are needed to be solved. We discuss these issues and ways to solve them in Section II. Then we present a detailed description of our proposed DCF in Section III. Experimental results and comparison with the IEEE 802.11 DCF performance are shown in Section IV. Finally in Section V, we consolidate our work and state our future research directions in this area.

II. CHALLENGES IN AD HOC OPERATIONS OF THE PROPOSED DCF

Several problems that are not present in access point based operations of our proposed DCF arise when we consider the ad hoc operation of it. In this section, we discuss and propose solution for each of these issues.

A. Sub-channel Assignment

In access point based operations, a device needs to associate itself with the AP. So, AP has a complete list of active devices and can assign sub-channels to them uniformly. In ad hoc operations, there is no such central control. But in that mode, devices require to transmit a short broadcast message named *network discovery message* (NDM) periodically so that each device can know its neighbors. We can use these NDMs to calculate a uniform sub-channel assignment in a distributed manner and thus solve the problem. So, in this work, we will assume that each node can calculate which sub-channel it will use to transmit its RTS message. When the medium is idle for more than DIFS period, a node can transmit an RTS using its assigned sub-channel or listen for the channel to decode transmissions from others. When a device receives and decodes an RTS that was intended for it, it halts its own backoff counter and waits for the completion of the current contention cycle. Then it transmits a CTS message using the sub-channel at which it has received that RTS message. In ad hoc mode, we cannot consolidate CTS and ACK messages as we have done in AP based operations because these CTS and ACK messages can be generated from multiple devices.

B. Receiver End Virtual Collisions

In access point based communications, all the uplink transmissions are intended for the access point. As we are considering a single cell network where all the devices can listen and

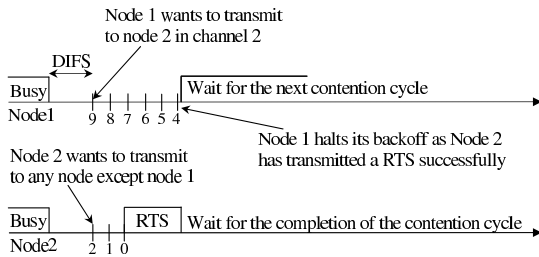


Fig. 2: Backoff halting due to receiver busy state.

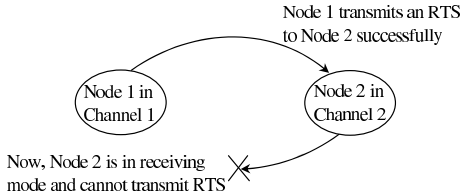


Fig. 3: A simple case of deadlock in the proposed DCF with two nodes in two sub-channels.

decode the transmission of other devices, a collision cannot occur at the receiver end. But in ad hoc operations, the receiver can be in other sub-channels than the transmitter. So, when a device is in midway of its RTS transmission, its intended receiver may also start his own RTS transmission, thus the first RTS message will be ignored by the receiver. Again, a device may already have started its RTS transmission when another device starts transmitting an RTS to it, resulting in the loss of that second RTS. We will call this phenomenon as the receiver end virtual collisions (though it is not a collision, one RTS message is lost because the receiver is in the transmitting mode and did not listen for that RTS message), which is illustrated in Fig. 1. As we increase the number of sub-channels, more time is required to transmit an RTS message and the probability of receiver end collisions increases, especially when the number of devices per sub-channel is low.

C. Backoff Halting

When a node is inside its backoff stage and senses that its intended receiver node has sent an RTS message successfully, it must halt its backoff (see Fig. 2). Because at that point, it will be meaningless to send that RTS message; it will eventually be neglected by the intended receiver device, which is already transmitting. Also, when in backoff, a node can successfully receive and decode an RTS message from other devices. At that point, it must freeze its backoff counter and reply with a CTS when the contention cycle is over. These two issues were not present when we discussed access point based operations, where a node will halt its backoff only when any other device in its own group has transmitted an RTS message.

D. Deadlock

The previous issue of backoff halting can cause a deadlock in our proposed DCF. Consider a very simple case - there are

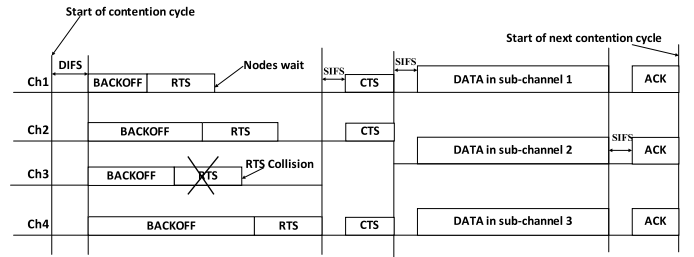


Fig. 4: The proposed DCF for ad hoc operations.

only two sub-channels and one device is assigned to each of them. When the first device transmits its RTS, it is intended towards the second device, which cannot transmit any RTS now (see Fig. 3). So our policy for AP based operations will not work here anymore, as it requires at least one RTS transmission from each sub-channel. To cope up with this new issue, we introduce a new parameter - $RTS_{timeout}$, after which period the contention cycle is over, no matter how many RTS has been sent. This $RTS_{timeout}$ will help us avoid the deadlock and has important impacts on the performance of the proposed DCF when the number of devices per sub-channel is very low.

In the next section, we will present the operation of our new DCF that overcomes all the challenges discussed above.

III. OUR PROPOSAL

In ad hoc operations of our proposed DCF, we assume that all the devices can find the sub-channel assigned in a distributed way, and each of them can try to send data to any other device randomly. We can describe our proposed DCF as follows:

- 1) Nodes periodically broadcast a NDM that contains information about the sub-channel it is using, along with other parameters.
- 2) When a device joins the system, it can transmit a *Hello* message. Other devices respond with NDM. The new device can then find the sub-channel that it will use by selecting the least used sub-channel. The whole procedure can be repeated periodically to accommodate topology changes in the network.
- 3) Nodes in the same group share the same frequency sub-channel and contend among themselves to capture the channel using the same backoff rules as in IEEE 802.11 DCF.
- 4) After transmitting RTS, the sender(s) wait for CTS. Other nodes in its group halt their backoff until the next contention cycle starts (by sensing the channel idle for more than DIFS period). So, there can be no more than one successful RTS sent from each group in a contention cycle.
- 5) When in backoff, nodes can listen for the channel. If a node can decode an RTS and decide that its intended receiver has transmitted that message, it halts its backoff and waits for the next contention cycle. Alternatively, if a device receives an RTS with its own address in the

destination address field, it halts its backoff counter and replies with a CTS message when the contention cycle is over.

- 6) The contention cycle ends as soon as either each group has sent one RTS (successful or unsuccessful), or a $RTS_{timeout}$ has occurred. No device will initiate a new RTS after the contention cycle is over, only the ongoing RTS transmissions will continue.
- 7) When the contention cycle is over, nodes that have received an RTS message wait for the channel to remain idle for SIFS time and then transmit CTS using the same sub-channel at which they have received that RTS.
- 8) Devices that have received CTS message, calculate sub-carrier assignment in a distributed ways. They start transmitting data using the calculated sub-channels after waiting for SIFS time. Devices can detect a collision by the absence of a CTS in response to its RTS. Then they wait for the beginning of another contention cycle using the same binary exponential backoff rules used by the IEEE 802.11 DCF.
- 9) Finally, after waiting for SIFS time, the receiving devices transmit ACK using the same sub-channels at which they have received data.

We can summarize the operation of our proposed DCF in Fig. 4, where we have divided the nodes into 4 groups. Nodes transmit RTS in their assigned sub-channel. Some RTS message may still collide as we can see in case of sub-channel 3. In Fig. 4, the contention cycle ended as each group has transmitted at least one RTS message. Nodes respond with CTS after waiting SIFS time using the same sub-channel at which they have received RTS. Based on the number of CTS message, the channel is divided into 3 sub-channels for transmitting data. ACK messages are transmitted using the same sub-channels at which the nodes have received their data.

IV. SIMULATION RESULTS

To analyze the performance of our proposed DCF, we developed a discrete event driven simulation using SimJava2 [13] and examined every aspect of our system. We assumed perfect channel conditions so that we can consider only the impact of our modifications. We tried to remain close to the IEEE 802.11 a/g protocol [14]. We did not use adaptive modulation and coding (AMC); rather we used fixed data rate of 36 Mbps to simulate both traditional IEEE 802.11 and our proposed system. We varied the number of nodes from 1 to 50 and experimented with 2, 4, and 8 sub-channels to examine the performance of our proposed system.

Fig. 5 shows the impact of our modifications on the number of RTS message per unit time. In all cases, number of RTS message in our proposed DCF is lower than that in the standard IEEE 802.11 DCF. When there are too few nodes per sub-channel (n_i), the main contributing factor is backoff halting due to receiver end transmissions as discussed in Section II-C. In those cases, $RTS_{timeout}$ occurs most of the time, rather than normal completion of the contention cycle. When the number of devices per sub-channel is moderately high (say,

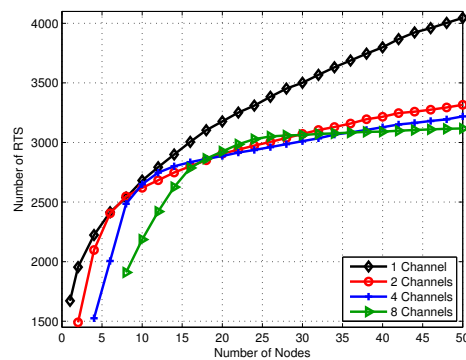


Fig. 5: Number of RTS message vs. number of nodes.

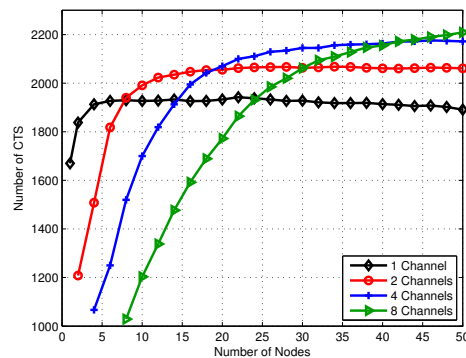


Fig. 6: Number of CTS message vs. number of nodes.

$n_i \geq 4$), less RTS messages are wasted in collisions and the number of successful RTS message increases. So, the total number of RTS message decreases as we increase c .

Fig. 6 shows that the number of CTS message per unit time increases as we increase the number of devices present in the system. But when n_i is too small (say, less than 4), the impact of receiver end virtual collisions and backoff halting due to busy receivers cause our DCF to perform lower than the IEEE 802.11 DCF. As the number of nodes per sub-channel increases, these two problems are reduced and our sub-channelized DCF performs better than the standard protocol. As in the case of AP based operations, here we can see that the number of CTS message increases when we increase the number of devices until it reaches its maximum and then remains almost constant as we are considering saturation load.

Percentage of time slots carrying payload data (also can be considered as data throughput) is directly related with the number of CTS message per unit time and we can see a similar trend in Fig. 7. We can see that traditional IEEE 802.11 DCF uses the channel to carry payload data for only 43% of the total time on an average, whereas we can reach up to 51% using our proposed DCF for ad hoc operations. Though this improvement is less dramatic than in the case of AP based operations of our proposed DCF due to the factors discussed in Section II, we can expect a throughput improvement of up to 20%.

Then we consider the percentage of RTS messages that

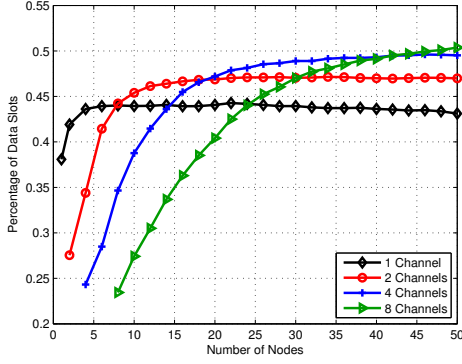


Fig. 7: Percentage of time the channel is used for carrying payload data vs. number of nodes.

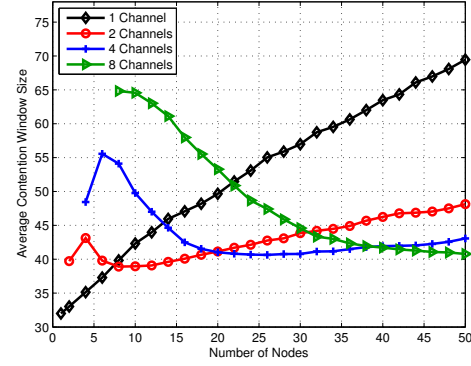


Fig. 9: Average contention window size vs. number of nodes.

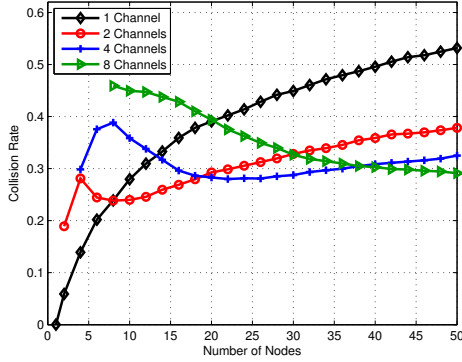


Fig. 8: RTS collision rate vs. number of nodes.

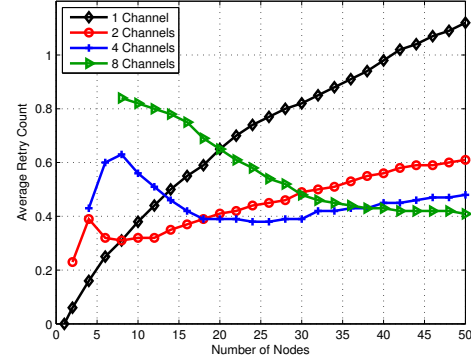


Fig. 10: Average retry count vs. number of nodes.

are lost due to collisions (Fig. 8). We can see that collisions in standard IEEE 802.11 DCF increase very quickly as the number of devices increases and it can reach up to 53% with 50 nodes present in the system. Sub-channelization can decrease collisions significantly, except for the case when the number of devices per sub-channel is too low. In those cases, the receiver end collisions discussed in Section II-B causes the RTS collision probability to be very high. As we increase c , each RTS message takes c times longer duration than in the standard protocol. So, the probability that the intended receiver starts transmission during this period will increase with increase in c . As n_i increases, this discrepancy is reduced and we can see that our proposed DCF can reduce collisions by up to 45%. Of course, the improvement is far less than that we have seen in the case of AP based operations of our DCF in [12].

The binary exponential backoff rule used in IEEE 802.11 DCF doubles the backoff window size when a collision is detected, until it reaches its maximum. So from Fig. 9, we can see that the average contention window size in standard IEEE 802.11 DCF is much higher than that in our proposed DCF, except for the anomaly that we have discussed above. Sub-channelization decreases collisions for $n_i \geq 4$ and we can see that the average contention window size decreases as we increase c . Similar trend can be seen for average retry count (Fig. 10). For example, when there are 40 devices

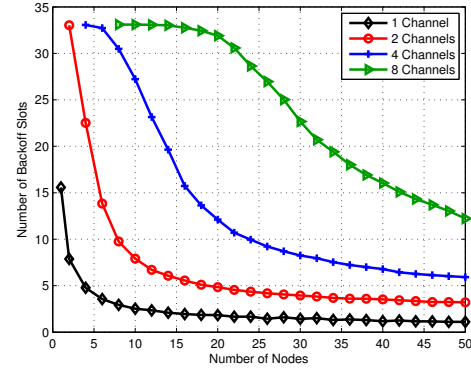


Fig. 11: Duration of backoff period/cycle vs. number of nodes.

contending to access the channel, the standard IEEE 802.11 DCF requires 2 RTS messages to successfully transmit a data packet, whereas our proposed DCF with 4 sub-channels requires only 1.4 RTS messages on average, thus saving available transmission time and power.

Let us consider the time wasted in backoff per contention cycle. From Fig. 11 we can see that the $RTS_{timeout}$ plays a crucial role in the performance of the protocol for lower values of n_i , where deadlocks are more probable. In our simulations, we have set this parameter to 33 time slots. So, we can see that the average duration of backoff is clipped at about 33 slot time, and it remains there until $n_i \geq 2$, then it gradually decreases.

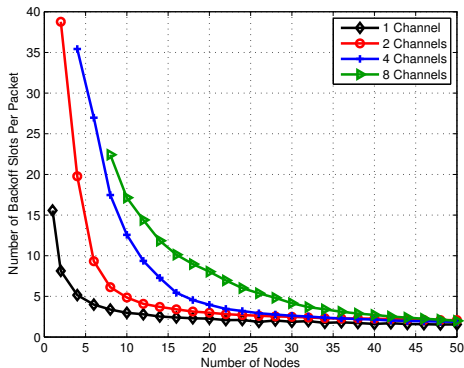


Fig. 12: Backoff slots wasted to transmit a successful RTS vs. number of nodes.

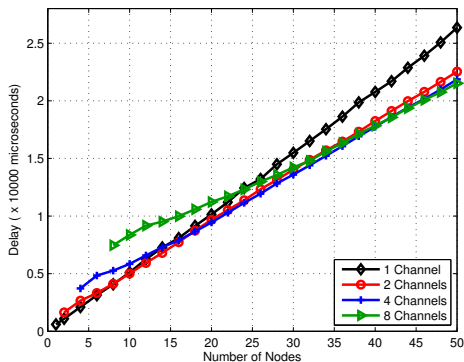


Fig. 13: Average transmission delay vs. number of nodes.

As our protocol requires at least one RTS transmission from each sub-channel or a $RTS_{timeout}$ to consider the end of a contention cycle, average duration of the backoff period is higher than that in the standard IEEE 802.11 DCF. But if we consider the amount of time wasted in backoff to transmit each RTS message successfully, we can find that the time wasted in our proposed DCF is very close to the standard protocol except for the case when n_i is too low (Fig. 12).

As our final performance measure, we consider average delay of packet transmission in Fig. 13. Delay is measured from the time a packet is in the head of queue (HoQ) of its corresponding device to the time it is successfully transmitted, including retransmissions. The same anomaly as in the previous cases is seen here. When there are too few nodes per sub-channel, our proposed DCF performs worse than the standard IEEE 802.11 DCF. But for higher values of n_i , we can see that the average transmission delay decreases as we introduce sub-channelization. Increasing c does not reduce delay much, as we have explained in Section II-B that the probability of receiver end virtual collisions increases when we increase c , specially for lower values of n_i . But yet, we can achieve up to 18% decrease in average delay with our proposed DCF.

From all these performance analysis and comparisons, we can conclude that though our proposed sub-channelized DCF does not perform as good as it performed for AP based

WLANs, it can reduce collisions by up to 45%, improve throughput by up to 20% and decrease delay by up to 18% when the number of devices per sub-channel is moderately high. From our experiments, we can see that when $n_i \leq 3$, our sub-channelized DCF performs lower than the standard IEEE 802.11 DCF. In that case, the embedded software can choose $c = 1$, and $RTS_{timeout} = \infty$ so that our protocol converges into the standard DCF, without requiring any hardware modifications. Based on the number of devices found by listening NDMs, a device can decide about using sub-channelization. Introducing adaptive modulation and coding can improve the performance of our DCF further and careful measures should be taken to utilize the full benefits of our proposed DCF.

V. CONCLUSION

In this paper, we have extended our proposed DCF [12] for ad hoc operations in a single cell wireless local area network. We have pointed the additional challenges that we had to face along with their remedies. We have shown that even for moderate number of users, our proposed DCF can overcome the overhead introduced by the protocol and can improve the performance of WLANs substantially. We are now working to develop a Markov chain model to find a complete analytical analysis of our proposed protocol for ad hoc operations. In that analysis, modeling the receiver end virtual collisions as well as backoff halting due to receiver busy state remain as a major research challenge.

REFERENCES

- [1] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535–547, Mar. 2000.
- [2] Y. Xiao and J. Rosdahl, "Throughput and delay limits of IEEE 802.11," *IEEE Commun. Lett.*, vol. 6, no. 8, pp. 355–357, Aug. 2002.
- [3] J. J. Garcia-Luna-Aceves, H. R. Sadjadpour, and Z. Wang, "Challenges: towards truly scalable ad hoc networks," in *Proc. MobiCom*, 2007.
- [4] S. Karande, Z. Wang, H. Sadjadpour, and J. J. Garcia-Luna-Aceves, "Optimal scaling of multicommodity flows in wireless ad hoc networks: Beyond the gupta-kumar barrier," in *Proc. MASS*, 2008, pp. 102–113.
- [5] M. Morelli, C.-C. J. Kuo, and M.-O. Pun, "Synchronization techniques for orthogonal frequency division multiple access (OFDMA): A tutorial review," *Proc. IEEE*, vol. 95, no. 7, pp. 1394–1427, Jul. 2007.
- [6] K.-D. Lee and V. C. M. Leung, "Fair allocation of subcarrier and power in an OFDMA wireless mesh network," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 11, pp. 2051–2060, Nov. 2006.
- [7] J. Gross, S. Valentin, H. Karl, and A. Wolisz, "A study on the impact of inband signaling and realistic channel knowledge for an example dynamic OFDM-FDMA system," *Euro. Trans. Telecommun.*, vol. 16, no. 1, pp. 37–49, 2005.
- [8] S. Valentin, T. Freitag, and H. Karl, "Integrating multiuser dynamic OFDMA into IEEE 802.11 WLANs - LLC/MAC extensions and system performance," in *Proc. IEEE ICC*, May 19–23, 2008, pp. 3328–3334.
- [9] H. S. Lichte, S. Valentin, F. Eitzen, M. Stege, C. Unger, and H. Karl, "Integrating multiuser dynamic ofdma into IEEE 802.11a and prototyping it on a real-time software-defined radio testbed," in *Proc. TridentCom*, May 21–23, 2007, pp. 1–9.
- [10] M. Veyseh, J. J. Garcia-Luna-Aceves, and H. R. Sadjadpour, "OFDMA based multiparty medium access control in wireless ad hoc networks," in *Proc. IEEE ICC*, Jun. 14–18, 2009, pp. 1–6.
- [11] M. Veyseh and J. J. Garcia-Luna-Aceves, "Parallel interaction medium access for wireless ad hoc networks," in *Proc. ICCCN*, Aug. 2008.
- [12] H. S. Ferdous and M. Murshed, "Enhanced IEEE 802.11 by integrating multiuser dynamic OFDMA," in *Proc. WTS*, Apr., 2010.
- [13] Sim java 2, "www.icsa.informatics.ed.ac.uk/research/groups/hase/simjava."
- [14] B. OHara and A. Petrick, *IEEE 802.11 handbook*, 2nd, Ed. IEEE Press.