

Research Article

G-DCF: Improving System Spectral Efficiency through Concurrent Transmissions in Wireless LANs

Ayinebyona Eliab ¹, Yonghwi Kim ¹, Joosang Lee ², Jeong-Gun Lee ¹
and Jungmin So ²

¹Smart Computing Laboratory, School of Software, Hallym University, Chuncheon, Republic of Korea

²Department of Computer Science and Engineering, Sogang University, Seoul, Republic of Korea

Correspondence should be addressed to Jungmin So; jso1@sogang.ac.kr

Received 30 May 2019; Revised 4 August 2019; Accepted 16 August 2019; Published 11 September 2019

Academic Editor: Alessandro Bazzi

Copyright © 2019 Ayinebyona Eliab et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents G-DCF, a MAC protocol for wireless LANs that can improve system spectral efficiency of wireless LANs by allowing more concurrent transmissions. The 802.11 DCF creates exposed terminals which are nodes that can transmit successfully but are blocked by carrier sensing. More potential exposed terminals are created when APs are densely placed, limiting spatial reuse of channels and thus system throughput. In order to allow concurrent transmissions from exposed terminals, G-DCF establishes groups in the network. Members of a group are nodes located within the carrier sense range of each other but can transmit packets concurrently. Whenever one member of a group wins the channel and transmits its packet, other nodes in the group also start transmission, triggered by the group ID included in the preamble. Contention window is adjusted according to the group size for fair share of the channel. Performance evaluations show that G-DCF can significantly improve system throughput and fairness over 802.11 DCF, especially when the APs are densely deployed.

1. Introduction

In IEEE 802.11 distributed coordination function (DCF), the exposed terminal problem is one of the factors that degrades system spectral efficiency [1]. When a node has a packet to transmit, it first listens to the channel to detect any ongoing transmissions. If the node senses a signal (or multiple signals) and the signal strength is higher than a predetermined threshold called *carrier sense threshold* (CST), the node decides the channel is busy and defers its transmission in order to avoid packet collision. CST is a tunable parameter, which decides how aggressively a node transmits its packets. If CST is low, a node will defer its transmission even at a very weak signal. Sometimes a node may unnecessarily defer its transmission even when the transmission will not cause a packet collision. This is called the *exposed terminal problem* because the sender is “exposed” to interference while its receiver is not.

Consider the scenario in Figure 1. Suppose AP_X is transmitting a packet to node A. If AP_Y concurrently

transmits its packet to B, its signal will become interference at node A. However, since node A is located far away from AP_Y , the packet can successfully be decoded in spite of the interference. Nevertheless, when AP_X is transmitting, AP_Y will sense the channel as busy and defer its transmission, resulting in waste of channel resource.

If AP_Y increases its CST, it may be able to transmit its packet simultaneously to AP_X . However, increasing CST aggravates the hidden terminal problem, which is also harmful for system performance. A hidden terminal is a node which senses the channel as idle but can cause packet collisions when it transmits a packet. In Figure 1, suppose AP_X is transmitting to node C instead of node A. In this case, if AP_Y transmits concurrently, node C will experience high interference and therefore fail to receive its packet. This scenario shows the limitation of adjusting CST to mitigate hidden and exposed terminal problems.

The main problem of carrier sensing-based medium access control is that while the channel condition at the receiver decides successful reception, the sender determines

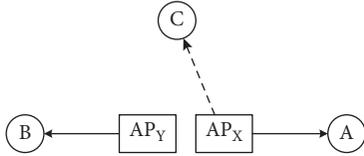


FIGURE 1: An example network scenario. Nodes A and C are associated with AP_X , while node B is associated with AP_Y .

whether to transmit or not based on the channel condition at its vicinity. AP_Y does not know where node B is, and how close it is located to the interferer. Moreover, AP_Y does not know if its signal will destroy the current transmission of AP_X at node A, which is possible if it is closely located to node A. Due to this lack of information, AP_X and AP_Y cannot transmit concurrently even though it is safe for both nodes to transmit.

In this paper, we propose an extension to 802.11 DCF called G-DCF (group DCF), where potential exposed terminals are grouped together so that they can transmit simultaneously and thus achieve higher system spectral efficiency. In G-DCF, nodes that can safely transmit concurrently are grouped together. If node A and node B are in the same group, they share a group ID. If node A wins the channel through contention and transmits a packet, it includes its group ID in the preamble. When node B receives the preamble and finds out that the group ID included in the preamble is the same as its group ID, the node can begin transmission concurrently, if the channel was idle before receiving A's preamble. Using simulations, we show that this simple idea can significantly improve system throughput, especially when the density of APs is high. Also, G-DCF also improves fairness over conventional 802.11 DCF by reducing the number of contenders from individual nodes to groups of nodes.

Rest of the paper is organized as follows: Section 2 discusses existing work that addresses the exposed terminal problem. Section 3 discusses the impact of exposed terminal problem on system throughput through a simple experiment and motivates the idea of grouping the nodes. Section 4 presents the details of the proposed protocol, G-DCF. Section 5 evaluates performance of G-DCF through extensive simulations, comparing with other existing protocols. Finally, Section 6 concludes the paper with remarks for future work.

2. Related Work

Improving system spectral efficiency has become an important issue in present day wireless LANs due to high traffic demand and increasing density of APs and mobile stations. The new wireless LAN standard IEEE 802.11ax particularly focuses on this issue and includes techniques such as BSS coloring [2]. The purpose of these techniques is to allow more concurrent transmissions in a given area, thereby increasing spatial reuse. The carrier sensing-based channel access method of IEEE 802.11 DCF was a great success and has been used in wireless LANs until this day, but its system spectral efficiency is significantly limited due to the hidden

and exposed terminal problems. In practical wireless LANs, the exposed terminal problem is shown to be more prevalent than the hidden terminal problem [3].

As discussed earlier, CST is an important parameter which decides the aggressiveness of nodes in terms of packet transmission. More hidden terminals are created if CST is higher, and more exposed terminals are created if CST is lower. Naturally, the choice of CST has a significant impact on the system performance [4]. Many proposals have been made in which the CST is dynamically adjusted to achieve the best performance [5, 6]. However, since alleviating one problem aggravates another, there is a limit to how much we can achieve by adjusting the CST (we show that through a simple experiment in the next section).

Another approach to address the exposed terminal problem (without aggravating the hidden terminal problem) is to identify potential exposed terminals and make them transmit concurrently even if they sense the channel busy. Vutukuru et al. [7] proposed to use CMAP (conflict MAP), which records information on whether a node should defer transmission when it overhears a packet traveling from a source to a destination. The CMAP is constructed by each node observing packet collisions and is passed to its one-hop neighbors. Whenever a node overhears a packet, it looks up the CMAP and decides whether it should defer its transmission or not. C-MAC [8] is a MAC protocol for wireless sensor networks (WSNs) that uses SNR measurements to find out whether two nodes can successfully transmit simultaneously. By measuring received signal strength from neighboring nodes at offline, nodes can estimate whether the packets can successfully be decoded at their receivers when multiple nodes transmit simultaneously. When a node wants to transmit a packet, it first overhears the ongoing transmission and finds out the sender, receiver, and the transmit power. Then, based on the offline measurements, the node decides whether it can safely send its packet concurrently with the ongoing packet.

CSMA/OCA [9] uses request to send (RTS) and clear to send (CTS) messages to identify hidden and exposed terminals. Before sending a packet, the node overhears RTS and CTS packets sent by other nodes. If no CTS was overheard and the destination of every RTS overheard is not a neighboring node, the node is in the exposed terminal position and can transmit its packet concurrently with the ongoing packet. Yao et al. [10] proposed IRMA, which addresses control packet collisions that can occur when exposed terminals transmit concurrently. While packets transmitted from two exposed terminals may successfully be received at their respective receivers, CTS and acknowledgment (ACK) frames sent back from the receivers may experience too much interference at the senders. In order to increase reliability of receiving those frames, IRMA encodes essential information into signatures which can be more reliably decoded in the presence of high interference. Wang et al. [11] proposed FAST, a system which uses full-duplex capability to address the exposed terminal problem. All senders, receivers, and victims (nodes interfered by the transmission) send "attachment" codes to signal their status. When a node wants

to transmit a packet, it collects the attachment codes from the neighborhood and decides whether its transmission will cause collisions.

Concurrent-MAC [12] uses a centralized approach to identify exposed terminals, where nodes measure signal strengths from neighbors and report to the central server, and the server computes which nodes can transmit concurrently. Whenever a node transmits, it selects one of its neighboring nodes as a “privileged node” who can transmit concurrently with the node and includes its ID in the header. When the privileged node overhears the header, it can begin transmission immediately. A dynamic sensitivity control (DSC) algorithm [13] was proposed to be incorporated into IEEE 802.11ax, where nodes independently and dynamically select CST based on the received signal strength index (RSSI) between the sender and the receiver. The main idea of DSC is that a mobile station located close to its access point (AP) can use a high CST because it can tolerate high interference at the receiver. On the contrary, a mobile station located far away from its AP should use a low CST in order to protect itself from packet collisions.

Other existing works on the exposed terminal problem include changing parameters other than the carrier sense threshold such as transmit power [14] and modulation and coding scheme (MCS) [15]. Also, special hardware can be used to address the problem, such as directional antenna [16].

The proposed protocol, G-DCF, adapts ideas from existing works such as C-MAC [8] and concurrent-MAC [12] in which concurrently transmittable nodes are identified through SNR measurements. However, unlike the other protocols, G-DCF removes the need to read the MAC header from packets on the fly, by precalculating groups and including group ID in the signal field of the preamble. Since the MAC header is often modulated using MCS for high data rate, it requires higher SNR to decode the MAC header and find out whether a concurrent transmission is possible. In contrast, the signal field of preamble is transmitted at a low data rate, making it more reliable for nodes to receive the information. Also, since preamble comes before the MAC header, a node can quickly find out whether it can concurrently transmit or not, compared to waiting for the MAC header. Compared to other existing works, G-DCF does not require special hardware such as directional antenna or full-duplex capability, and special control messages such as RTS and CTS.

3. Preliminaries

The main idea of G-DCF is to group nodes that are potentially “exposed” to each other, so they can transmit together as a group. In this section, we discuss whether the idea of grouping nodes can potentially improve the spatial reuse of a wireless LAN system. We have implemented a simple simulator using Python, which simulates a network environment shown in Figure 2. In the scenario, APs are placed in a grid shape and mobile stations are randomly deployed in the area. Once deployed, each mobile node associates with the nearest AP. We assume all mobile stations have

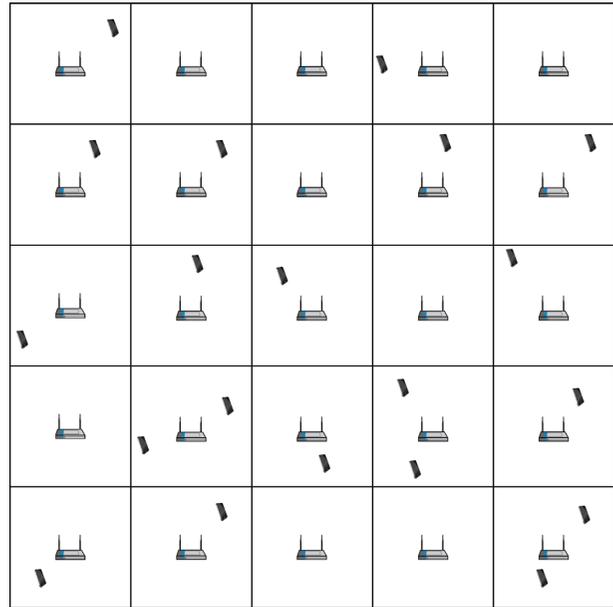


FIGURE 2: Environment for the simulation experiment. APs are placed in a grid shape, and mobile stations are randomly placed in a square-shaped area.

downlink traffic to receive from their APs, so APs constantly try to win access to the channel and send their packets. We are interested in how many packets can be successfully received at their receivers, when the packets are transmitted concurrently.

Under this environment, we model the behavior of 802.11 DCF as follows. Time is divided into “rounds.” In each round, a mobile station is randomly chosen, and its associated AP transmits a packet to the mobile station. Once the AP transmits a packet, other APs that are in the vicinity of the transmitter will sense the channel busy, if the signal strength of the ongoing transmission is higher than CST. APs that sense the channel busy are “blocked” cannot concurrently transmit in that particular round. If there remain APs that are not blocked yet, one of the mobile stations that are associated with those APs is randomly selected, so that the corresponding AP can transmit concurrently with the ongoing transmission. This process is repeated, until there is no AP left that can transmit concurrently in that round. Once the set of transmitters are decided, we determine whether their transmissions will successfully be received at their receivers. We count the number of packets successfully received in each round.

We assume that all nodes transmit packets at a transmit power of 20 dBm. The received signal strength is calculated using the log-distance path loss model, which uses equation (1). In the equation, the reference distance (d_0) is 1 m and the reference path loss (PL_0) is 46.67 dBm. The path loss exponent (γ) is 3, and the random variable that models fading (X_g) is 0. These parameter values are adopted from default parameters used in the ns-3 simulator [17]:

$$P_r = P_t - \left(PL_0 + 10\gamma \log_{10} \frac{d}{d_0} + X_g \right). \quad (1)$$

At each receiver, SNR is calculated using equation (2). In the equation, the noise floor (N_f) is -93.97 dBm, which is also a value used in the ns-3 simulator. We decide the packet is successfully received if the SNR is larger than 23 dB, which is an SNR value required to decode a packet sent at 54 Mbps data rate:

$$\text{SNR} = \frac{P_s}{\sum P_1 + N_f}. \quad (2)$$

We also model a protocol called ‘‘Ideal,’’ where nodes magically know whether they can concurrently transmit with the ongoing transmission or not. In each round, a mobile station is randomly chosen, and its associated AP transmits a packet to the mobile station. Once the AP transmits a packet, only the nodes that cannot transmit together with the ongoing transmission are blocked. So in the Ideal protocol, exposed terminals are not blocked by the transmission, whereas the hidden terminals are blocked. Then, among unblocked nodes, another transmission is randomly chosen and added to the list of concurrent transmissions in that round.

In the experiment, we measured the average number of successful transmission per round for DCF (with various CSTs) and the Ideal protocol. We also measured Jain’s fairness index [18] in order to see how the channel resource is shared among the nodes. We have varied the number of APs, while fixing the size of simulation area at $80 \text{ m} \times 80 \text{ m}$ and the number of mobile stations at 20. We ran each simulation for 1000 rounds, and each point in the graph is an average of 100 runs with different topology and random number generator seed. The result is shown in Figure 3.

First, we look at 802.11 DCF. When CST is -92 dBm, the average number of concurrent transmissions is 1 regardless of the number of APs. This means that when one node transmits, the entire area is blocked by the carrier sensing. When CST is increased to -82 dBm and -72 dBm, the number of concurrent transmissions increases due to increased spatial reuse. The number of concurrent transmissions also increases with the number of APs, but after a certain point, it remains the same. This means that even if the APs are densely deployed, the throughput of DCF is limited by the CST. When the CST becomes even higher, the transmissions are affected by hidden terminals. The number of concurrent transmissions is smaller when CST is -52 dBm compared to when CST is -62 dBm because hidden terminals cause packet collisions. Compared with DCF, the Ideal protocol knows the exposed and hidden terminals can send more concurrent packets successfully. From Figure 3(a), it seems that the performance of Ideal is similar to that of DCF with CST equal to -62 dBm. However, from Figure 3(b), it can be observed that the fairness of DCF with CST = -62 dBm is much lower than Ideal. This is because for DCF with CST = -62 dBm, only the nodes that are very close to the destination can send packets, whereas other nodes fail to transmit due to hidden terminals. On the contrary, the fairness of Ideal is very high, meaning that nodes that are far away from their destinations have more chance of sending their packets successfully. In summary, if the hidden

terminals and exposed terminals are known, the system can achieve higher throughput and fairness compared to the original 802.11 DCF.

The proposed protocol, G-DCF, models the Ideal protocol by identifying exposed terminals through signal strength measurements. In order to avoid hidden terminals, a low CST is used. Then, the identified exposed terminals are grouped together so that they can transmit their packets together. Note that, in the above experiment, we did not model the behavior of backoff countdown in DCF. In practice, simultaneous transmissions could happen due to multiple nodes finishing their backoff at the same time. With more nodes in the network, this ‘‘backoff collision’’ could occur more frequently, affecting the system performance. In order to model protocol behaviors more accurately, we use the ns-3 simulator [17] to conduct performance evaluations in Section 5. In the next section, we describe the details of G-DCF.

4. Proposed Protocol: G-DCF

4.1. Group Formation. We assume that a group of APs deployed in a region collaborates with each other and can exchange messages through wired interface. There could be a central coordinator, or one of the APs could become the leader and coordinate the entire group of APs. When a mobile station is associated with one of the APs, two group IDs are assigned for this mobile station: one for downlink and one for uplink. Initially, all group IDs are zeroes meaning that the DL and UL of mobile stations do not belong to any group.

In order to operate G-DCF, each mobile station maintains RSSI of neighboring APs, including its associated AP. Whenever a mobile station receives a beacon, it updates the RSSI of the AP by calculating the exponential moving average (EMA) of the RSSI. Also, each mobile station keeps track of how much uplink traffic it has to send to the AP. The traffic information is required because a node does not need to be in a group if it does not have any downlink or uplink traffic. When a mobile station associates with an AP, it sends the RSSI and the traffic information to the AP. When the APs receive the information from the mobile stations, they forward the information to the coordinator. Also, each AP sends how much downlink traffic it has for each associated mobile station. There are various ways of measuring traffic demand, but we use the ratio of time packets that are backlogged in the queue. For example, suppose the measure period is 60 seconds. If AP_X has a packet destined for node C in its transmit queue for 30 seconds in the 60-second period, the downlink traffic demand of node C is 50%. Also, if node A has a packet in its transmit queue for 45 seconds during that 60-second period, the uplink traffic demand of node A is 70%. If the traffic demand exceeds a certain threshold, the downlink or uplink of a mobile station could be included in a group. The coordinator maintains tables similar to Figures 4(a) and 4(b), in order to manage the groups. Note that, in real implementation, node labels such as A, B, C, AP_X , and AP_Y are replaced by MAC addresses. In the mobile station table,

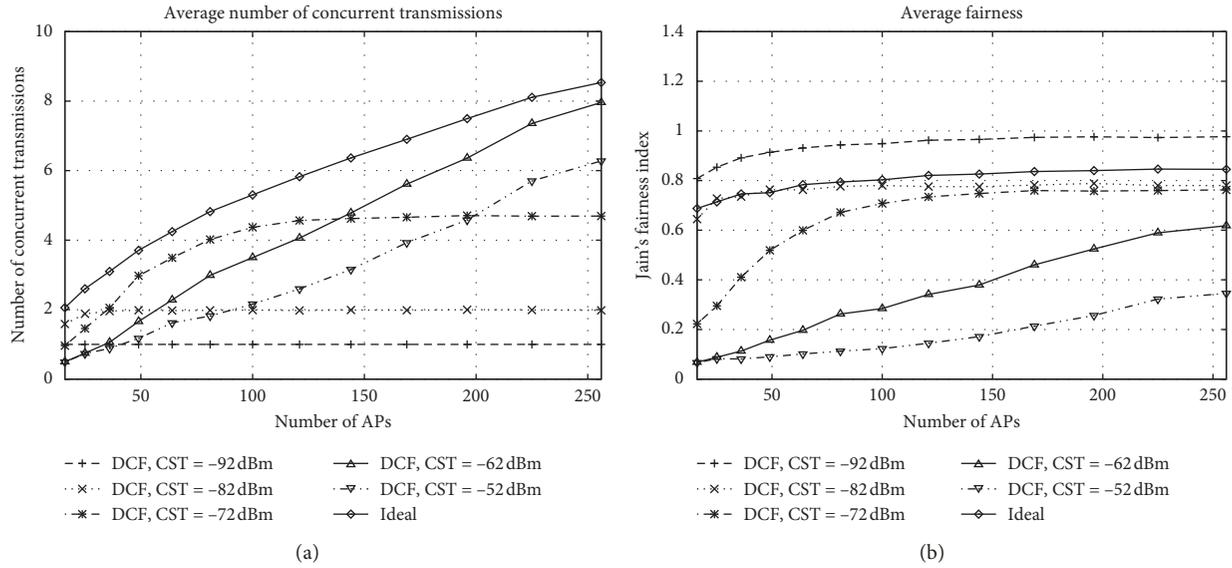


FIGURE 3: Average number of concurrent transmissions and fairness varying number of APs. (a) Throughput. (b) Fairness.

Address	Assoc.	RSSI (dBm)		DL		UL	
		AP _X	AP _Y	Group	Traffic	Group	Traffic
A	AP _X	-50.0	-79.0	0	1	0	0
B	AP _Y	-77.0	-46.0	0	1	0	0
C	AP _X	-52.0	-61.0	0	1	0	0

(a)

RSSI	AP _X	AP _Y
AP _X	—	-62.0
AP _Y	-62.0	—

(b)

FIGURE 4: Tables maintained at the coordinator records the information of each (a) mobile station table and (b) AP RSSI table.

the value of “traffic” field is 0 if the traffic demand is less than the threshold and 1 if the traffic demand is higher than the threshold. The AP RSSI table records signal strength of beacons between the APs.

Once the coordinator gathers information from the APs, it starts calculating the groups. The algorithm for establishing groups is a greedy-style algorithm, in which a node tries to join the “best” group for DL and UL. The algorithm is illustrated using a flow diagram in Figure 5. Note that DL and UL of a mobile station are considered separately; we can think of DL and UL of a mobile station as two different mobile stations. Also, only mobile stations with high traffic demand are considered in group formation. If a DL or a UL of a mobile station has no traffic, its group ID becomes zero. In the flow diagram of Figure 5, the mobile stations considered in the algorithms are stations with high traffic demand.

For clarity, let us assume all mobile stations only have downlink traffic demand. The coordinator randomly picks a mobile station i , in order to see if node i can join another group. To do that, the coordinator picks another mobile station j . Now, the coordinator checks the following six conditions before deciding to include node i in the same group as node j :

- (1) If node i and node j are in the same Basic Service Set (BSS) (associated with the same AP), they cannot be in the same group.
- (2) If node i and node j are already in the same group, then no work needs to be done.
- (3) If the senders of node i and node j (APs in the downlink case) are outside the carrier sensing range, then they do not need to be in the same group.
- (4) If node i is already in a group, then we compare the size of i 's group (S_i) and the size of j 's group (S_j). If $S_i > S_j + 1$, then i does not move to j 's group.
- (5) If $S_i = S_j + 1$, we compare the minimum SNR of nodes for the case where i stays in its group and the case where i moves to j 's group. If the minimum SNR is not increased, then i does not move to j 's group.
- (6) If node i joins j 's group, the nodes in j 's group should be able to concurrently transmit together. If this is verified through SNR calculation, node i joins j 's group.

The first condition is obvious because two mobile stations connected to the same AP cannot transmit or receive packets concurrently. For the second condition, node i and

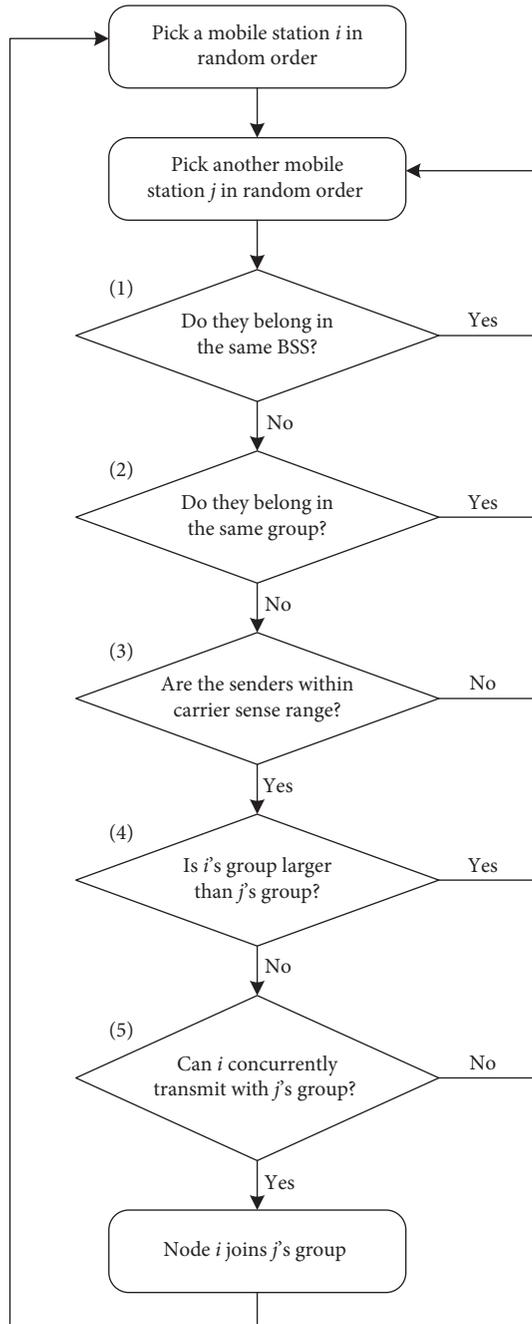


FIGURE 5: Flow diagram of the group formation algorithm.

node j are considered to be in a separate group if their group ID is zero. Other than zero, if their group ID is the same, it means they are already in the same group.

The third condition says that if the senders are not within the carrier sensing range, they are not grouped together. For example, in Figure 1, we were considering to group the DL of nodes A and B. But suppose AP_X and AP_Y are outside the carrier sense range, meaning that when AP_X transmits, still AP_Y finds the channel idle. Then, they are not exposed terminals, and they could transmit together even without grouping. So we do not gain benefit from grouping the DL of nodes A and B. Also, as will be described in the next section, concurrent transmission is triggered by a preamble. If the

transmitters are too far away, the preamble might not reach the transmitters in the same group and thus may not trigger concurrent transmission.

We have noted that DL and UL of a mobile station are treated independently when establishing the groups. It is possible that a group contains multiple DL connections as well as UL connections. While for other conditions, there is no difference in handling DL and UL, and checking the third condition needs extra information for UL connections. In order to check whether the two transmitters are within the carrier sense range of each other, we need to have RSSI information between the transmitters. If we are checking to see if two DL connections can be grouped together, we can use the AP RSSI table to check whether the transmitters are within the carrier sense range. Also, if we are checking to see if a DL connection and a UL connection can be grouped together, we can use the mobile station table to check if the transmitters are within the carrier sense range (assuming the link is symmetric.) However, if we are trying to see whether two UL connections can be grouped together, we need to know the RSSI between two mobile stations. RSSI of APs is easier to measure and maintain because APs periodically transmit beacons, but it is more difficult to obtain RSSI between mobile stations. There are two ways to address the problem. First, nodes can build an additional table which contains RSSI of neighboring mobile stations. The RSSI of neighboring mobile stations are obtained opportunistically by overhearing their transmissions. Second, if the RSSI of mobile stations is not available, we can limit a UL to be always checked against a DL, when the coordinator tries to group two links together. In other words, at least one transmitter must be an AP. In that case, we can use the mobile station table to check the third condition. Although two UL connections cannot be directly checked for group formation, they can eventually be in the same group if a DL connection is also in the group.

The fourth condition is a greedy-style optimization, which tries to make larger groups. Consider the scenario in Figure 6. There, the DL of nodes A, B, and C are in group 1 (black), whereas the DL of nodes D, E, and F are in group 2 (red). The sizes of the two groups are both 3. If node D moves from group 2 to group 1, the size of group 1 will become 4, which is larger than 3. In this case, node D moves to group 1 if the coordinator thinks all 4 nodes in group 1 can successfully transmit together. Group 2 is now with two nodes, but there may be other nodes without a group that can join group 2 because joining a group of two is easier than joining a group of three.

The fifth condition is a further optimization in addition to the fourth condition. Consider the scenario in Figure 7. In the figure, the DL of nodes A, B, and C is in group 1, whereas the DL of nodes D and E is in group 2. If node B leaves group 1 and moves to group 2, the group sizes will remain the same. In this case, we compare the minimum SNR of any node in the two groups. In the case of Figure 7, when node B is in group 1, the minimum SNR is 24 dB which is experienced by node A. If node B moves to group 2, the minimum SNR becomes 29 dB of node B. In this case, we allow node B to

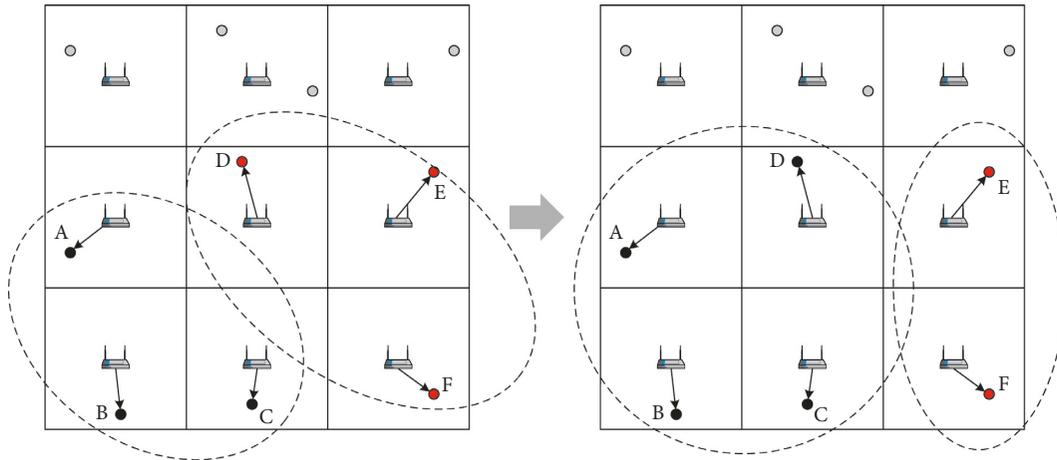


FIGURE 6: Group formation algorithm: nodes A, B, and C are in group 1 and nodes D, E, and F are in group 2. If node D can be moved to group 1, then it is preferred to do so because it will create a larger group.

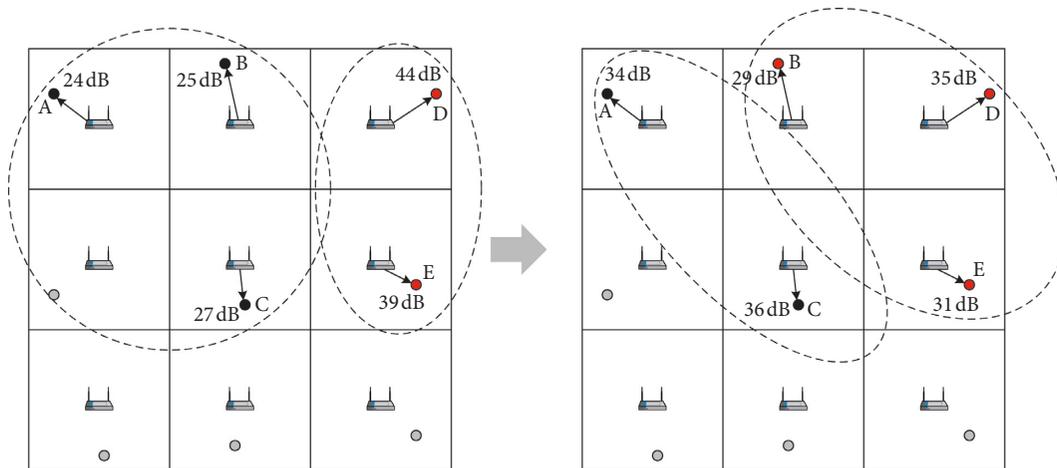


FIGURE 7: Group formation algorithm: nodes A, B, and C are in group 1 and nodes D and E are in group 2. If node B moves to group 2, the group sizes will remain the same but the minimum SNR is increased from 24 dB to 29 dB. In that case, it is preferred that node B moves to group 2.

move because with higher SNR there is higher chance that an additional node can join the group.

Finally, if node i is ready to join node j 's group, the coordinator has to verify if all nodes in j 's group can transmit concurrently. That is, if the nodes in j 's group transmit simultaneously, the SNR at each receiver must be larger than a threshold. The SNR at each receiver is calculated using equation (2). Then, the estimated SNR is compared with SNR_{TH} , which is calculated as in equation (3). In the equation, SNR_{min} is the minimum SNR required to successfully decode the packet modulated using a certain modulation and coding scheme. SNR_{TH} adds a margin M to SNR_{min} in order to account for variations in channel conditions:

$$SNR_{TH} = SNR_{min} + M. \quad (3)$$

If the final condition is satisfied, node i moves to node j 's group. If j 's group ID is zero, the coordinator assigns a new group ID.

Once the coordinator finishes the group assignment, it sends the assignment results to all the APs. The APs maintain DL group IDs for all its associated mobile stations and send the UL group IDs to the mobile stations. Since nodes may join, leave, or change positions, the SNR measurements can become stale over time. Also, the traffic demand of mobile stations may change over time as well. In order to reflect the changes, the coordinator should recalculate the groups occasionally, based on the updated information. As described earlier, each mobile station keeps track of the RSSI of its associated AP. Also, each mobile station stores the RSSI value when the node was assigned the latest group ID. If the difference between the current RSSI and the stored RSSI exceeds a certain threshold, the node sends a message to its AP reporting the change of RSSI (the report could be included in the data or ACK message sent from the mobile station to the AP). The AP forwards this information to the coordinator, and the coordinator recalculates the groups triggered by the

message. The coordinator may recalculate groups for the entire network or just reset the group ID for the particular mobile station and try to assign the node to a new group. The group assignment is also triggered whenever a node joins or leaves the network, or traffic demand of a node changes.

4.2. Protocol Operation. Consider the scenario in Figure 8. Mobile stations A, B, and C are associated with AP_X, AP_Y, and AP_Z, respectively. Suppose the coordinator has assigned group ID 1 to the DL of node A and node B and assigned group ID 2 to the DL of node C. Similar to IEEE 802.11 DCF, the nodes follow the random backoff procedure to gain access to the channel. Suppose AP_X finishes counting down its backoff counter and wins the channel. Then, AP_X begins transmission by sending a preamble. In G-DCF, the transmitter includes its group ID in the physical layer convergence protocol (PLCP) header. This is similar to the BSS coloring scheme of IEEE 802.11ax, where BSS color (a unique ID assigned to the BSS) is included in the PLCP header [2].

When AP_X transmits the preamble, it will be heard by AP_Y and AP_Z. Normally in DCF, when the neighboring nodes receive a preamble, they will sense the channel busy, set up a network allocation vector (NAV), and pause the backoff counting. However, in G-DCF, AP_Y reads the preamble and checks the group ID, which is 1 in this case. Since transmission from AP_Y to node B is also assigned group 1, it means AP_Y can transmit concurrently with the ongoing transmission. If the channel was idle before receiving the preamble, AP_Y stops the backoff and immediately starts its transmission towards node B. If there were other transmitters that were assigned group 1, they would also start the transmission if the channel was idle before receiving the preamble. It is as if AP_X's preamble is a trigger for all nodes in the group to start transmission. On the contrary, AP_Z will also receive the preamble sent by AP_X. Upon checking the group ID, it finds out that the group ID is different from its assigned group ID. Then, AP_Z behaves similar to DCF, where the node sets up an NAV and pauses the backoff. The behavior of G-DCF is illustrated in Figure 9.

While the uplink traffic only has a single destination, the downlink traffic may have multiple destinations. In G-DCF, different group IDs are assigned to the DL of different mobile stations, even if they are associated with the same AP. In other words, an AP may have multiple group IDs for DL, depending on the destination. In Figure 8, suppose AP_X started transmitting, AP_Y received the preamble to find out that the group ID was 1. Suppose the DL of node B was assigned group 1, the DL of node D which is another mobile station associated with AP_Y was assigned group 2. If AP_Y was preparing to send a packet to node D, it cannot transmit together with AP_X since their group IDs are different. So AP_Y loses the chance to transmit concurrently with AP_X, even though it can transmit concurrently if the destination was node B.

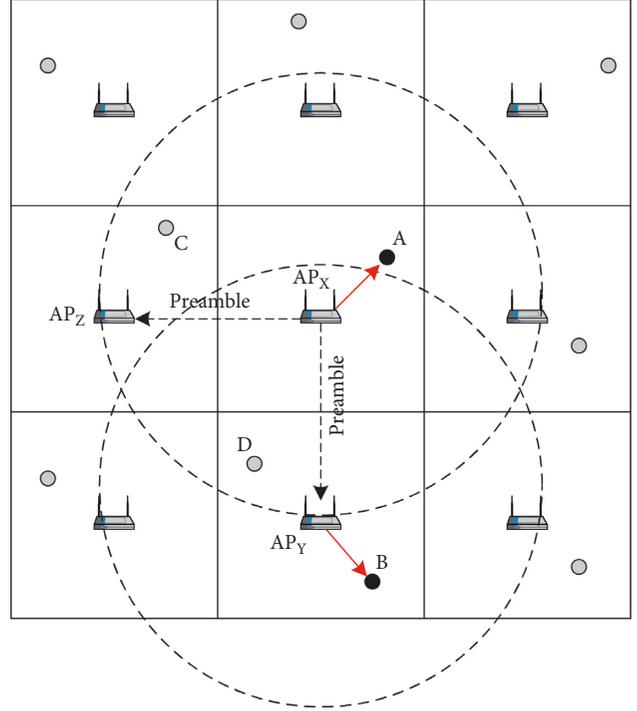


FIGURE 8: An example scenario where G-DCF is used. The DL of nodes A and B is in the group 1, whereas the DL of node C is in group 2.

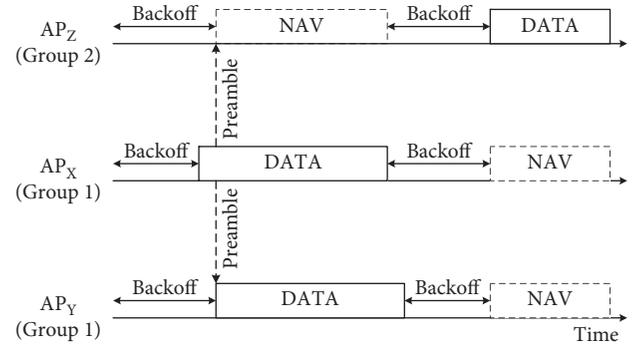


FIGURE 9: Behavior of G-DCF. When AP_X transmits a packet to node A, it includes group ID 1 in the preamble. When AP_Y receives the preamble, it can immediately transmit a packet to node B if the channel was idle before receiving the preamble. When AP_Z receives the packet, it sets up the NAV and defers its transmission.

In order to maximize system spectral efficiency, G-DCF allows an AP to switch the order of transmissions if the packet can be transmitted concurrently with ongoing transmissions. In the above example, suppose AP_Y was waiting to send a packet to node D. However, AP_Y also has a packet to send to node B in its transmit queue. If AP_Y receives a preamble which includes group ID 1, then it immediately starts sending a packet to node B.

Suppose AP_Y successfully transmits a packet to B and receives an ACK. In DCF, the node will reset the contention window and pick a random number from the contention

window to contend for the channel again. However, in this example, the backoff counter of AP_Y is not zero because the transmission was triggered by a preamble and not by finishing the backoff. Suppose AP_Y resets the backoff counter and chooses a random number from the contention window. Then, while waiting for the channel, the same event could occur where a preamble from a node in group 1 is received and AP_Y sends a packet to node B again. If this happens repeatedly, the packet to node D can be delayed without limit. Thus, in G-DCF, the backoff counter and the contention window is not altered if the transmission was not from the packet in the front of the transmit queue. In the example, suppose the backoff counter was 5 when AP_Y received a preamble and started transmitting to node B. After the transmission, the backoff counting is resumed at 5. On the contrary, if AP_Y transmits a packet to node D which was at the front of the transmit queue, the contention window and the backoff counter are managed similar to DCF, after the transmission.

4.3. Contention Window Adjustment. Once multiple nodes are grouped together, one node will potentially transmit a packet when any member of the same group wins the channel. Because of that, nodes belonging to a large group will have a higher chance of accessing the channel than nodes in a small group, which results in unfair share of channel bandwidth. One way of improving fairness is to adjust the contention window size according to the group size.

After transmitting a packet, a node picks a random number in the range $[0, CW - 1]$, where CW is the contention window size. Initially, CW is set to CW_{\min} . Whenever a transmission fails, CW is doubled until it reaches CW_{\max} (CW_{\min} and CW_{\max} are predefined system parameters). Whenever a transmission succeeds, CW is reset to CW_{\min} .

Suppose a node does not belong to any group. Then, if CW is N , the node has to backoff for $(N - 1)/2$ time slots in average. We would like to adjust CW of the nodes so that even when the node is in a group of size m , the expected number of backoff slots is the same as a node without a group. In the following, we calculate the contention window size for a group of m nodes so that the expected number of backoff slots becomes the same regardless of the group size. Note that a mobile station can be assigned two group IDs, one for DL and one for UL. To avoid confusion, let us assume that all traffic we have is downlink for this calculation, so a mobile station i is assigned a single group ID. In practice, DL and UL are separately considered, and the same calculation can be applied regardless of whether the node has DL or UL traffic.

We define X_j as the number randomly chosen by node j from the contention window. If there are multiple nodes in a group, the number of slots for backoff is the minimum of random numbers chosen by the nodes. Thus, the probability that the number of backoff slots is i is $P\{\min X_j = i\}$ and can be calculated as follows:

$$\begin{aligned} P\{\min X_j = i\} &= P\{\min X_j \geq i\} - P\{\min X_j \geq (i + 1)\} \\ &= P\{X_1 \geq i\}P\{X_2 \geq i\} \cdots P\{X_m \geq i\} \\ &\quad - P\{X_1 \geq (i + 1)\} \cdots P\{X_m \geq (i + 1)\} \\ &= \left(\frac{N - i}{N}\right)^m - \left(\frac{N - (i + 1)}{N}\right)^m, \end{aligned} \quad (4)$$

where m is the number of nodes in the group and N is the contention window size. The expected number of backoff slots E_{X_j} is

$$\begin{aligned} E_{X_j} &= \sum_{i=0}^{N-1} i \cdot \left[\left(\frac{N - i}{N}\right)^m - \left(\frac{N - (i + 1)}{N}\right)^m \right] \\ &= 1 \times \left[\left(\frac{N - 1}{N}\right)^m - \left(\frac{N - 2}{N}\right)^m \right] + 2 \\ &\quad \times \left[\left(\frac{N - 2}{N}\right)^m - \left(\frac{N - 3}{N}\right)^m \right] + \cdots + (N - 1) \times \left(\frac{1}{N}\right)^m \\ &= \sum_{i=1}^{N-1} \left(\frac{i}{N}\right)^m = \frac{1}{N^m} \sum_{i=1}^{N-1} i^m. \end{aligned} \quad (5)$$

Suppose we define N_1 as the contention window size of an independent node. The expected number of backoff slots is

$$E_{X_j} = \frac{N_1 - 1}{2}. \quad (6)$$

So we would like to select N as the contention window size of nodes in the group of size m according to the following equation:

$$\frac{1}{N^m} \sum_{i=1}^{N-1} i^m = \frac{N_1 - 1}{2}. \quad (7)$$

This equation cannot be simply solved due to m . Instead, we approximate the left-hand side, which is in the form of Bernoulli numbers, into a simpler form as follows. We start with the equation which contains the sum of i^m :

$$\begin{aligned} N^{m+1} - 1^{m+1} &= \sum_{j=1}^{m+1} \binom{m+1}{j} \sum_{i=1}^{N-1} i^{m+1-j} \\ &= \binom{m+1}{1} \sum_{i=1}^{N-1} i^m + \binom{m+1}{2} \sum_{i=1}^{N-1} i^{m-1} + \epsilon, \end{aligned} \quad (8)$$

where $\binom{m+1}{j}$ is the number of j combinations from a set of $m + 1$ elements.

Assuming $m \geq 3$, we define ϵ as

$$\epsilon = \binom{m+1}{3} \sum_{i=1}^{N-1} i^{m-2} + \cdots + \binom{m+1}{m+1} \sum_{i=1}^{N-1} i^0. \quad (9)$$

Note that if m is 1 or 2, equation (7) can be easily solved. Rearranging equation (8), we get

$$\sum_{i=1}^{N-1} i^m = \frac{N^{m+1} - 1 - \epsilon}{m+1} - \frac{m}{2} \sum_{i=1}^{N-1} i^{m-1}. \quad (10)$$

Dividing both sides by N^m , we get

$$\frac{1}{N^m} \sum_{i=1}^{N-1} i^m = \frac{N}{m+1} - \frac{m}{2 \cdot N^m} \sum_{i=1}^{N-1} i^{m-1} - \beta, \quad (11)$$

where

$$\beta = \frac{1 + \epsilon}{(m+1) \cdot N^m}. \quad (12)$$

Note that if $m \geq 3$ and $N \geq 16$, β is a small number that can be neglected. In a similar way, we derive an equation for the sum of i^{m-1} :

$$\begin{aligned} N^m - 1^m &= \sum_{j=1}^m \left(\binom{m}{j} \sum_{i=1}^{N-1} i^{m-j} \right) \\ &= \binom{m}{1} \sum_{i=1}^{N-1} i^{m-1} + \epsilon'. \end{aligned} \quad (13)$$

This time, we only leave the first term of the sum and regard everything else as ϵ' . Then,

$$\frac{1}{N^m} \sum_{i=1}^{N-1} i^{m-1} = \frac{1}{m} - \beta', \quad (14)$$

where

$$\beta' = \frac{1 + \epsilon'}{m \cdot N^m}, \quad (15)$$

in which β' is a small number which we will ignore. So from equations (11) and (14),

$$\frac{1}{N^m} \sum_{i=1}^{N-1} i^m \approx \frac{N}{m+1} - \frac{1}{2}. \quad (16)$$

Finally, we apply equation (7) to obtain the contention window size N for a given m :

$$N = \frac{m+1}{2} \cdot N_1. \quad (17)$$

Whenever a node draws a random number from the contention window, the contention window is scaled according to equation (17). The procedure of increasing and decreasing contention window is the same as in original DCF. In the evaluations, we evaluate both cases where CW adjustment is applied and not applied to G-DCF.

5. Performance Evaluation

5.1. Simulation Setup. Performance evaluation of G-DCF was conducted using the ns-3 simulator [17]. The simulation environment consists of APs placed in a grid structure, and mobile stations randomly deployed throughout the whole area, as shown in Figure 2. Once deployed, each mobile station associates with the nearest AP. Each mobile station is

assigned with either a downlink or an uplink UDP traffic, which starts once the simulation begins.

For simulation parameters, we have used the 802.11a standard with 20 MHz channels. Data packets are sent at 54 Mbps rate, whereas control packets such as ACKs are sent at 24 Mbps. For path loss, we use the log-distance model with parameters described earlier in Section 3. A UDP packet is 1472 bytes, which becomes 1500 bytes after adding UDP and IP header. Unless otherwise specified, we deploy 100 APs in a 80 m \times 80 m area, and 20 mobile nodes are randomly placed inside the area. We assume that the mobile nodes always have UL or DL traffic, which makes them nodes with high traffic demand. The carrier sense threshold is -82 dBm, and CW_{\min} and CW_{\max} are 16 and 1024, respectively. For G-DCF, we have modified the PLCP header to include an 8-bit field called "Transmit Group ID (txgid)." The PLCP header of 802.11a is 48 bits. Thus, the PLCP header becomes 56 bits for G-DCF, which is an additional overhead for the protocol. To calculate the SNR threshold in equation (3), we use 23 dB as SNR_{\min} and 2 dB as the margin M . With these set of default values, we have varied the number of nodes, number of APs, simulation area size, and carrier sense threshold in order to create various network environments.

For performance metric, we have used the aggregate system throughput and Jain's fairness index [18]. Measuring fairness is important because often protocols achieve high throughput by letting a small subset of nodes dominate the channel, which is an undesired behavior. We have compared four different protocols. The first one is "802.11 DCF," which is the original DCF protocol. The second protocol is the dynamic sensitivity control protocol [13] proposed for IEEE 802.11ax standard. The protocol dynamically adjusts the carrier sense threshold based on the RSSI between AP and the mobile station. The carrier sense threshold is calculated according to equation (18), where C_i is the carrier sense threshold used for mobile station i and $RSSI_i$ is the RSSI of the associated AP measured at node i . For parameters, we use Margin = 25 dB, $CST_{\max} = -62$ dBm, and $CST_{\min} = -82$ dBm as suggested in the paper:

$$C_i = \max(C_{\min}, \min(C_{\max}, RSSI_i - \text{Margin})). \quad (18)$$

The third and fourth protocols are G-DCF. We measure the throughput and fairness of G-DCF with and without the contention window adjustment described in Section 4.3. With contention window adjustment, fairness among nodes is expected to improve but the total throughput will decrease by giving more chance to small groups and individual nodes. By comparing the two protocols, we study how the contention window adjustment scheme impacts the throughput and fairness of the system.

5.2. Results. In the first experiment, we have varied the number of mobile stations. We have deployed 100 APs in an 80 m \times 80 m area, and 5 to 100 mobile stations are randomly placed in the area. The carrier sense threshold was fixed at -82 dBm. The result is shown in Figure 10. The first observation to make is that the total throughput increases with the number of nodes for all protocols, as shown in

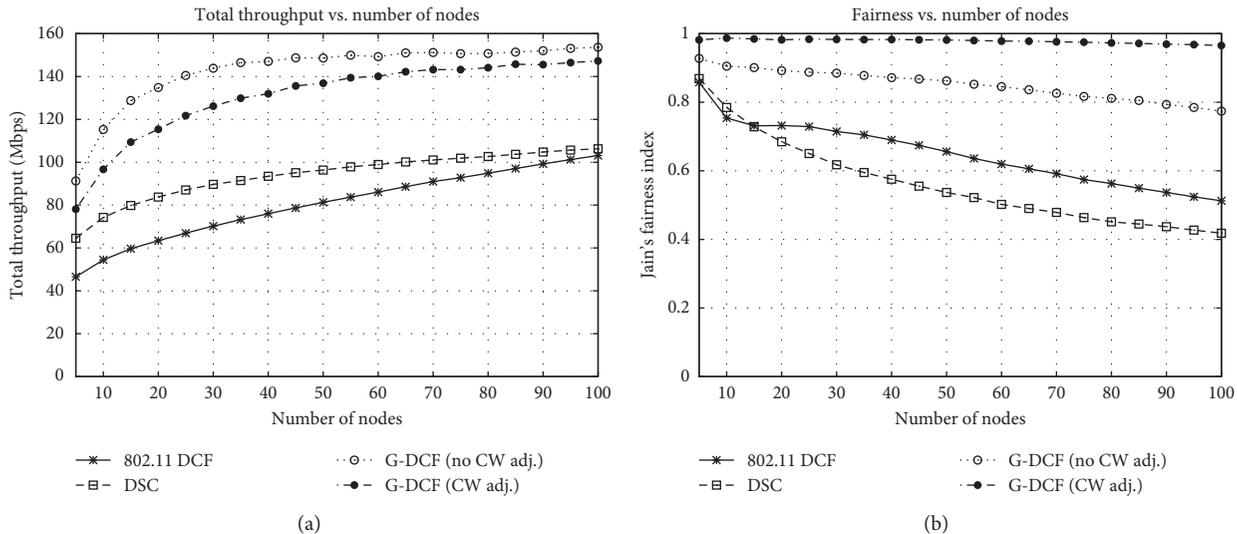


FIGURE 10: (a) Total throughput and (b) Jain's fairness index varying the number of mobile nodes.

Figure 10(a). This is because with more nodes, the chance that multiple nodes transmit concurrently is higher. Two nodes can transmit together if they are outside the carrier sense range of each other, or they happen to finish backoff at the same time. With more stations, the probabilities of both events are higher.

When two nodes transmit simultaneously, transmissions may fail due to low SNR at the receivers. This is possible when two closely located nodes accidentally finish backoff at the same time, or the nodes are hidden terminals to each other even when they are outside the carrier sense range. It is also possible that one of the packets is successfully received at the receiver, while the other packet is dropped due to low SNR. This happens when one node is close to its destination, while the other node is relatively far away from its destination. Suppose node A and node B transmitted concurrently but only node A successfully sent its packet to its destination. In that case, node B will double its contention window, while node A resets the contention window to the minimum value. Thus, it becomes more difficult for node B to access the channel. If node A is a hidden terminal to node B, then node B will very scarcely get a chance to send a packet because it constantly loses contention to node A. This is the major reason of unfairness in the CSMA/CA-based protocols, and that is why the fairness decreases as the number of nodes increases, as shown in Figure 10(b).

Comparing the protocols, we can see that DSC achieves higher throughput compared to 802.11 DCF, but its fairness is lower than DCF. If a node closer to the destination sets CST higher, these nodes will naturally have more chance of accessing the channel. Moreover, since these nodes transmit aggressively, more often they could become hidden terminals to nodes that are far away from their destinations. So we can say that DSC is achieving more throughput by giving more priority to mobile stations placed near the APs, which is not desirable in terms of fairness. For DSC, the throughput could become higher if

we increase CST_{max} to a value higher than -62 dBm. However, it will create more hidden terminals, and the fairness will decrease further.

On the contrary, G-DCF improves both throughput and fairness compared to 802.11 DCF. G-DCF achieves 1.5 to 2 times the throughput of 802.11 DCF, depending on the density of mobile stations. However, the fairness of 802.11 DCF drops significantly as the number of nodes increases, whereas the fairness of G-DCF is still very high. With contention window adjustment, the throughput improvement is achieved because more nodes transmit concurrently due to colliding backoff counters and hidden terminals. Since this behavior favors the nodes close to the destination, fairness is significantly degraded. This behavior occurs much less frequently in G-DCF because nodes are grouped together. Suppose a node is in a position where a hidden terminal always interferes with the node and prevents the node from successfully transmitting its packet. In G-DCF, this node could be grouped with other nodes so that it can transmit together with the other nodes in the group. The hidden terminal which interfered with the node in 802.11 DCF can now be blocked by the other nodes in the group. Also, since node groups contend for the channel instead of individual nodes, the number of contenders is smaller in G-DCF compared to 802.11 DCF. Thus, G-DCF shows higher throughput and fairness compared to 802.11 DCF, which is desirable for practical use.

In the second experiment, we varied the number of APs. Fixing the area size, we increased the number of APs from 4 to 256. The other parameters are the same as previous experiment ($80\text{ m} \times 80\text{ m}$ area, 20 nodes, $CST = -82$ dBm). The result is shown in Figure 11. Intuitively, the system throughput increases with increasing AP density. High AP density means the average distance between a mobile station and its associated AP is shorter. If the average distance is small, there is a higher probability that the transmission is

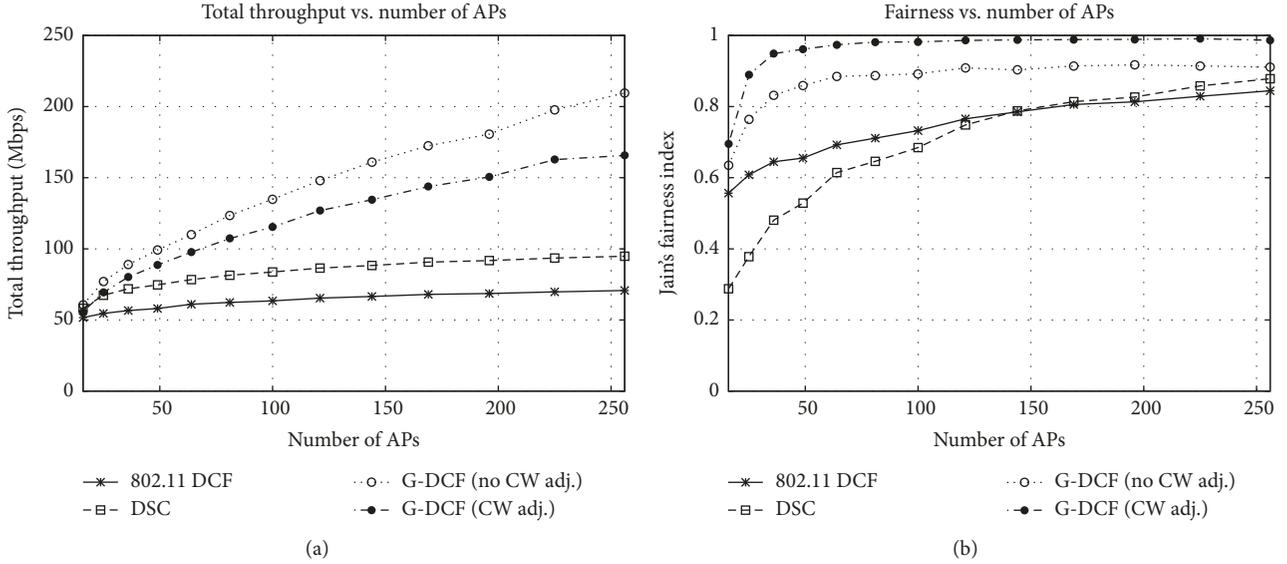


FIGURE 11: (a) Total throughput and (b) Jain's fairness index varying the number of APs.

successful when the nodes transmit concurrently. For G-DCF, more nodes can be grouped together because the SNR at the receiver is higher with more APs. For example, with 256 APs, the throughput can go up to 210 Mbps, which is equivalent to seven nodes transmitting concurrently, in average. (When there is a single pair of nodes, the UDP throughput is approximately 30 Mbps with saturated traffic.) With 256 APs, G-DCF (without contention window adjustment) could achieve 3 times the throughput of 802.11 DCF, with higher fairness index. On the contrary, DSC achieves 1.4 times the throughput of 802.11 DCF, with similar level of fairness. The fairness also increases with density of APs because when multiple nodes accidentally transmit due to collisions in backoff, there is higher chance that both transmissions will still survive. In other words, not many nodes suffer starvation due to frequent collisions, which is why the system achieves good fairness.

In the third experiment, we varied the size of simulation area from 1600 m^2 ($40 \text{ m} \times 40 \text{ m}$) to 160000 m^2 ($400 \text{ m} \times 400 \text{ m}$), while fixing the number of APs at 100. 20 mobile nodes are randomly deployed in the area, and the carrier sense threshold is fixed at -82 dBm . The result is shown in Figure 12. When area size is small, G-DCF achieves higher throughput and fairness compared to 802.11 DCF and DSC. However, as the area size becomes larger, the difference in both throughput and fairness diminishes. This is because when the network is sparse, not many exposed terminals exist in the network. When the network is very sparse, many pairs of nodes are outside carrier sense range, so they are not grouped together. Thus, the benefit of G-DCF diminishes when the network becomes sparse. For DSC, since the average RSSI of APs decreases with decreasing network density, the carrier sense threshold of nodes becomes lower, near CST_{\min} which is -82 dBm . Thus, the performance of DSC also becomes similar to that of 802.11 DCF. From this experiment, we can see that the benefit of G-DCF is higher when the network is dense.

In Figure 12(b), we can see that the fairness of 802.11 DCF shows a different behavior compared to other protocols. Specifically, the fairness of 802.11 DCF is especially low when the area size is $5000\text{--}15000 \text{ m}^2$. It is because that is the point when the area becomes slightly larger than the carrier sense range. When the simulation area is smaller than 5000 m^2 , almost the entire area is blocked by a transmission. But if the area size becomes larger, nodes at the edge of the area can transmit more often because they are outside the carrier sense range of each other. On the contrary, nodes at the center of the area have less chance of transmission because they have to contend with all nodes at the edge of the area. This is the reason the fairness is low. Once the area size becomes larger, the privilege of nodes at the edge of the area becomes smaller because now the nodes at the center are also outside the carrier sense range of the edge nodes. This behavior is not shown in G-DCF because nodes at the center can be grouped with nodes at the edge. If these nodes transmit together as a group, there is no privilege for the edge nodes because the nodes at the other edge will be blocked by the nodes at the center of the area.

In the next experiment, we varied the carrier sense threshold from -94 dBm to -44 dBm , whereas the other parameters are fixed at their default values ($80 \text{ m} \times 80 \text{ m}$ area, 100 APs, and 20 nodes). For this experiment, we have compared 802.11 DCF and G-DCF with and without contention window adjustment because DSC does not use a fixed carrier sense threshold. The result is shown in Figure 13. The throughput and fairness of G-DCF is higher than 802.11 DCF when the CST is low, but their performance becomes similar when the CST is high. Since the group formation algorithm of G-DCF only groups nodes that are within the carrier sense range, not many groups are established when CST is high because most nodes are outside of each other's carrier sense range. If no groups are formed, the performance of G-DCF becomes the same as 802.11 DCF. From this result, we can conclude that it is a

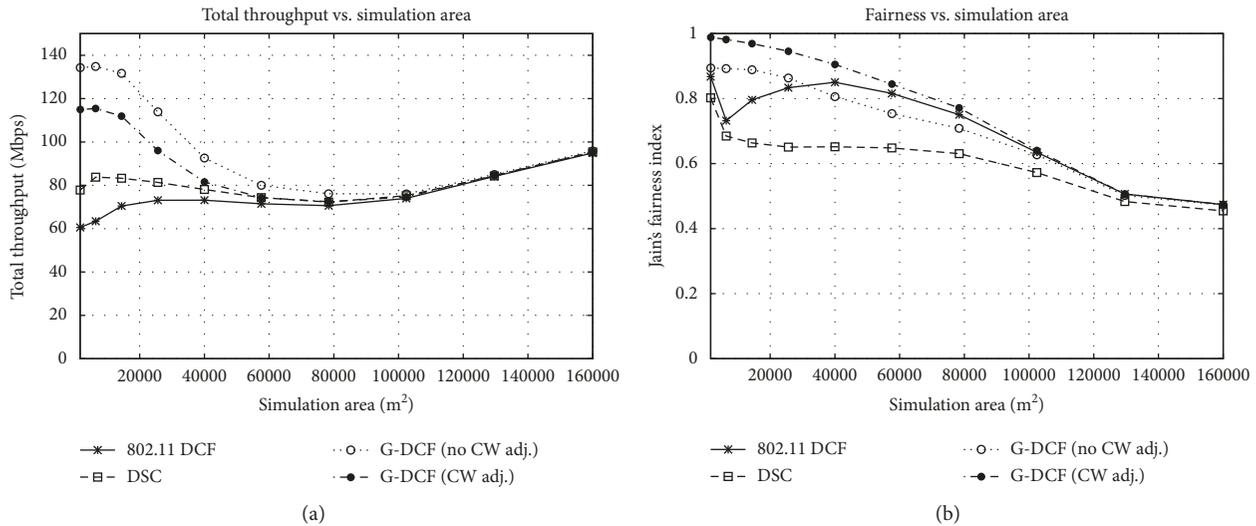


FIGURE 12: (a) Total throughput and (b) Jain's fairness index varying size of the simulation area.

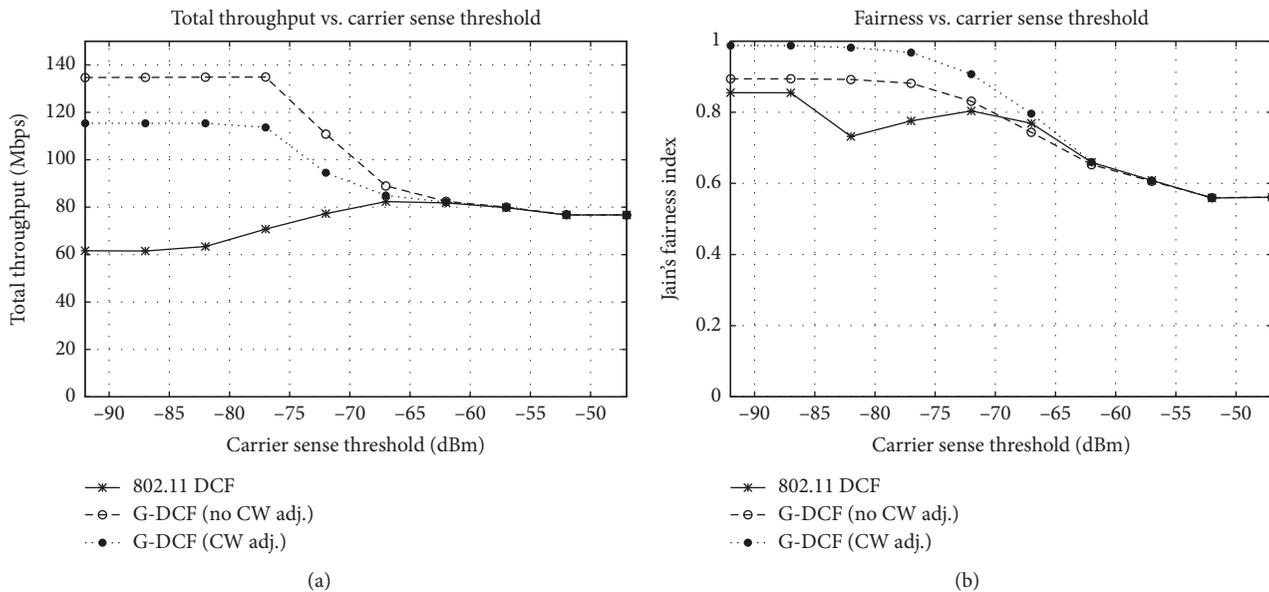


FIGURE 13: (a) Total throughput and (b) Jain's fairness index varying carrier sense threshold.

good practice to use G-DCF to allow concurrent transmissions for high spectral efficiency, while keeping CST low in order to prevent hidden terminals and thus improve fairness of the system.

In all the experiments until now, the mobile stations were stationary and did not move. If the mobile stations move around, concurrent transmissions within a group may no longer succeed. As discussed earlier, a mobile station reports to its AP when the RSSI of its associated AP changes by a certain threshold value. The coordinator triggers a group update when it receives this message forwarded by the AP. In the final experiment, we made the mobile stations move in a random direction at a speed of 3.6 km/h, which is similar to the speed of a pedestrian. We have varied the

threshold for group update from 1 dB to 16 dB. More group updates will be triggered with a low threshold, which would increase overhead but also would quickly reflect the changes in topology. All other parameters were fixed at their default values (80 m × 80 m area, 100 APs, and 20 nodes). The result is shown in Figure 14.

In Figure 14(a), we have compared the throughput of 802.11 DCF and G-DCF when there is mobility and also measured the throughput of G-DCF when there is no mobility. When the threshold is low, the throughput of G-DCF with mobility is similar to that when there is no mobility. Since the groups are updated quickly, not much throughput is lost due to failed concurrent transmissions. Note that, in some cases, the system achieves higher

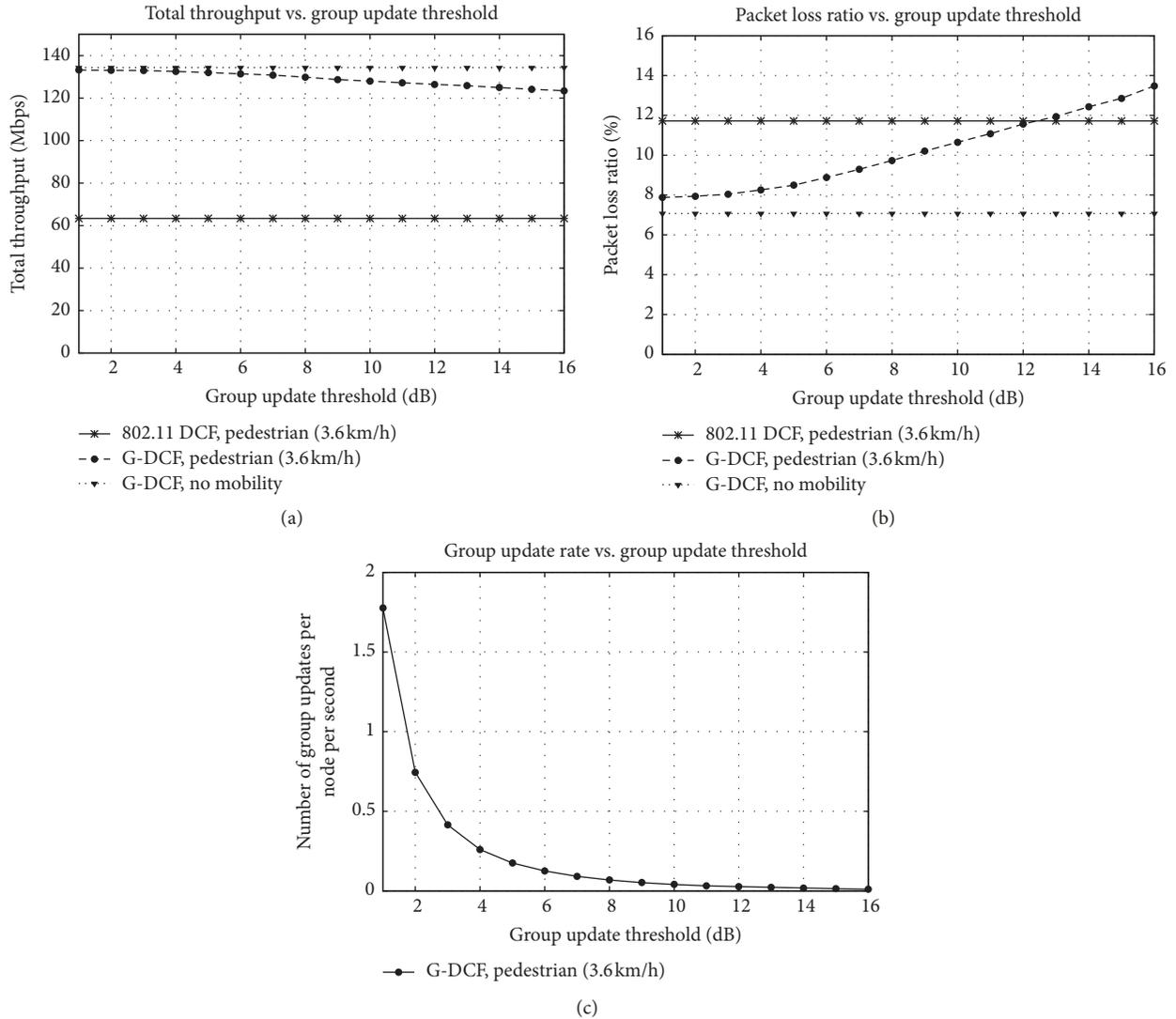


FIGURE 14: Results with mobile station mobility. The mobile stations move in the random direction at a speed of 3.6 km/h. (a) Throughput. (b) Packet loss ratio. (c) Group update rate.

throughput with mobility because the nodes move into better positions for concurrent transmissions. When the threshold becomes higher, some throughput is lost because the groups are not updated frequently and concurrent transmissions fail due to low SNR. Still, G-DCF achieves significantly higher throughput compared to 802.11 DCF, which does not allow concurrent transmission within the carrier sense range. Transmission failures due to outdated groups are reflected in Figure 14(b), which shows the packet loss ratio. The packet loss ratio of G-DCF increases with the threshold because nodes in a group may no longer be able to transmit concurrently once they move out of place. Then, the packet losses occur until the coordinator reassigns groups to the mobile stations. In Figure 14(c), the group update rate is measured. For example, when the threshold is 1 dB, each node reports a change of RSSI 1.8 times in a second. With 20 mobile stations, group updates are triggered 36 times in a second. If we assume the RSSI report packets are of the same size as the data packets (1500 bytes), the overhead for

sending RSSI updates is approximately 0.4 Mbps. Although this amount of overhead is not so significant, the overhead increases with the number of nodes, and too frequent group updates could also burden the coordinator and the links between APs and the coordinator. The group update rate quickly drops down if we increase the threshold. For example, the number of group updates is 0.4 when the threshold is 3 dB and 0.13 when the threshold is 6 dB. It could be a better practice to set the threshold which does not trigger the group updates too frequently, but frequently enough to reflect the changes in topology.

Figure 15 shows the throughput over time, when the mobile stations move in pedestrian speed. It shows that the total throughput changes over time due to mobility of the stations. For example, the system achieves a very high throughput between 20 and 22 seconds because the nodes are positioned so that more concurrent transmissions are possible. When the threshold is 20 dB, the group update rarely occurs, and we can see that the system achieves lower

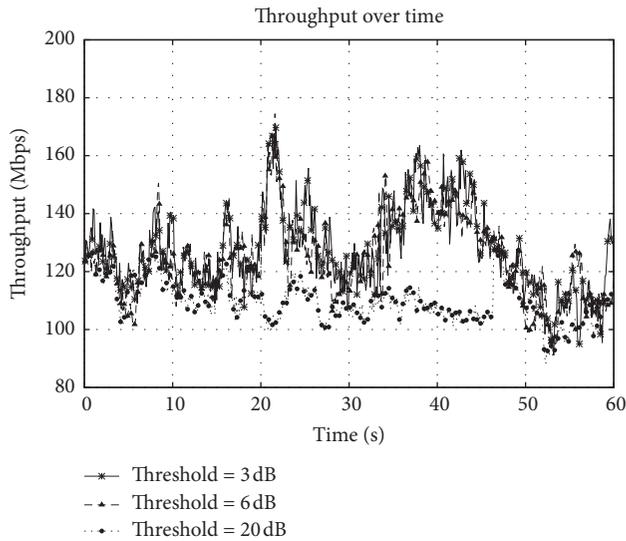


FIGURE 15: Throughput over time for G-DCF with different threshold for triggering group updates.

throughput compared to the case with lower thresholds. Especially, from 20 to 45 seconds, the groups are not updated properly to reflect the newest topology and therefore does not achieve high throughput. The throughput patterns of 3 dB and 6 dB are similar, but we can observe that, for several occasions, throughput of 3 dB is higher than throughput of 6 dB, such as 25 to 27 seconds and 42 to 44 seconds. The higher throughput is achieved by quickly adapting to changes in the network topology.

In summary, G-DCF can achieve higher throughput and fairness compared to 802.11 DCF and DSC, by grouping potential exposed terminals and allowing them to transmit concurrently. With contention window adjustment, almost maximum fairness can be achieved at the cost of a slight degradation in system throughput. G-DCF works best when the density of AP is high, which is a practical scenario with the cost of APs becoming cheaper.

6. Conclusion

The exposed terminal problem is a major factor that limits system spectral efficiency of wireless LANs. Exposed terminals are created from the fact that carrier sensing is done at the sender, whereas the channel status at the receiver is important for successful delivery of packets. In 802.11 DCF, nodes do not know whether they can transmit concurrently or not because they do not know the channel status at the receiver. Even if the nodes know which nodes can transmit concurrently, the MAC protocol should support concurrent transmission efficiently.

In this paper, we proposed G-DCF, a MAC protocol for wireless LANs which improves system throughput over 802.11 DCF by allowing exposed terminals to transmit concurrently. The main idea is to establish groups in the network and have all nodes in the same group transmit together. To form groups, mobile stations measure RSSI of the neighboring APs and report the information to the APs.

A coordinator gathers the information and makes groups based on the group formation algorithm. At the end of the algorithm, each mobile station is assigned two group IDs, one for downlink and one for uplink. In order to enable concurrent transmission, the group ID is included in the signal field of the preamble. Whenever a node receives a preamble from another node, it compares the group ID in the preamble and the group ID of itself. If the node has the same group ID with the preamble sender, the node immediately starts transmitting its packet. If the node is not in the same group with the preamble sender, the node sets NAV and defers its transmission, similar to 802.11 DCF.

Performance evaluation shows that G-DCF substantially improves system throughput compared to 802.11 DCF and DSC, a protocol which uses dynamic CST control. G-DCF also achieves higher fairness compared to 802.11 DCF, by reducing the number of contenders to groups instead of individual nodes. Also, nodes that suffered starvation because of hidden terminals can avoid starvation in G-DCF by getting into groups with other nodes. In summary, we believe G-DCF is a practical protocol that can be used to improve system spectral efficiency of wireless LANs. In the future, we plan to implement G-DCF on a FPGA-based software-defined radio (SDR) test bed in order to study the feasibility and real-life performance of the protocol.

Data Availability

The data used to support the finding of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Research Foundation of Korea under Grants NRF-2016R1D1A1B03930289 and NRF-2019R1A2C1005881.

References

- [1] K. Nishide, H. Kubo, R. Shinkuma, and T. Takahashi, "Detecting hidden and exposed terminal problems in densely deployed wireless networks," *IEEE Transactions on Wireless Communications*, vol. 11, no. 11, pp. 3841–3849, 2012.
- [2] B. Bellalta, "IEEE 802.11ax: high-efficiency WLANs," *IEEE Wireless Communications*, vol. 23, no. 1, pp. 38–46, 2016.
- [3] G. Judd and P. Steenkiste, "Understanding link-level 802.11 behavior: replacing convention with measurements," in *Proceedings of the 3rd International ICST Conference on Wireless Internet*, Austin, TX, USA, October 2007.
- [4] X. Yang and N. Vaidya, "On physical carrier sensing in wireless ad hoc networks," in *Proceedings of the IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, Miami, FL, USA, March 2005.
- [5] X. Zhang, H. Zhu, and G. Qiu, "Optimal physical carrier sensing to defend against exposed terminal problem in wireless ad hoc networks," in *Proceedings of the 23rd*

- International Conference on Computer Communication and Networks (ICCCN)*, Shanghai, China, August 2014.
- [6] J. Zhu, X. Guo, L. Lily Yang, W. Steven Conner, S. Roy, and M. M. Hazra, "Adapting physical carrier sensing to maximize spatial reuse in 802.11 mesh networks," *Wireless Communications and Mobile Computing*, vol. 4, no. 8, pp. 933–946, 2004.
 - [7] M. Vutukuru, K. Jamieson, and H. Balakrishnan, "Harnessing exposed terminals in wireless networks," in *Proceedings of the USENIX NSDI*, San Francisco, CA, USA, April 2008.
 - [8] M. Sha, G. Xing, G. Zhou, S. Liu, and X. Wang, "C-MAC: model-driven concurrent medium access control for wireless sensor networks," in *Proceedings of the IEEE INFOCOM 2009—the 28th Conference on Computer Communications*, Rio De Janeiro, Brazil, April 2009.
 - [9] S. Chakraborty, S. Nandi, and S. Chattopadhyay, "Alleviating hidden and exposed nodes in high-throughput wireless mesh networks," *IEEE Transactions on Wireless Communications*, vol. 15, no. 2, pp. 928–937, 2016.
 - [10] J. Yao, T. Xiong, J. Zhang, and W. Lou, "On eliminating the exposed terminal problem using signature detection," *IEEE Transactions on Mobile Computing*, vol. 15, no. 8, pp. 2034–2047, 2016.
 - [11] L. Wang, K. Wu, and M. Hamdi, "Combating hidden and exposed terminal problems in wireless networks," *IEEE Transactions on Wireless Communications*, vol. 11, no. 11, pp. 4204–4213, 2012.
 - [12] G. Hosseinabadi and N. Vaidya, "Concurrent-MAC: increasing concurrent transmissions in dense wireless LANs," in *Proceedings of the International Conference on Computing, Networking and Communications (ICNC)*, Kauai, HI, USA, February 2016.
 - [13] I. Selinis, M. Filo, S. Vahid, J. Rodriguez, and R. Tafazolli, "Evaluation of the DSC algorithm and the BSS color scheme in dense cellular-like IEEE 802.11ax deployments," in *Proceedings of the IEEE 27th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Valencia, Spain, September 2016.
 - [14] C. Huang, C.-T. Lea, and A. K.-S. Wong, "A joint solution for the hidden and exposed terminal problems in CSMA/CA wireless networks," *Computer Networks*, vol. 56, no. 14, pp. 3261–3273, 2012.
 - [15] J. Huang, G. Xing, and G. Zhou, "Unleashing exposed terminals in enterprise WLANs: a rate adaptation approach," in *Proceedings of the IEEE INFOCOM 2014—IEEE Conference on Computer Communications*, Toronto, Canada, April 2014.
 - [16] K. Adere and G. R. Murthy, "Solving the hidden and exposed terminal problems using directional-antenna based MAC protocol for wireless sensor networks," in *Proceedings of the International Conference on Wireless and Optical Communications Networks (WOCN)*, Colombo, Sri Lanka, September 2010.
 - [17] The NS-3 simulator, <https://www.nsnam.org/>.
 - [18] R. Jain, D.-M. Chiu, and W. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," Tech. Rep. TR-301, Digital Equipment Corporation (DEC), Maynard, MA, USA, 1984.



Hindawi

Submit your manuscripts at
www.hindawi.com

