Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.18576/amis/140515

Distributed List Hub Polling and Light Robust Super-Poll MAC Protocols for WLAN

Mahmoud Abd El-sattar *, *Nagwa M. Omar and Hosny M. Ibrahim*

Information Technology Department, Faculty of Computers and Information, Assiut University, Assiut, Egypt

Received: 22 Oct. 2019, Revised: 12 Jun. 2020, Accepted: 22 Jun. 2020
Published online: 1 Sep. 2020

Abstract: This paper presents two hub polling medium access control protocols for wireless local area networks based on the robust super poll protocol. The proposed protocols decrease the overhead and increase the throughput through eliminating broadcasting the polling list every super frame and eliminating the use of the chaining mechanism that is utilized in the robust super poll protocol in which all the remaining polling list is appended to every data frame that is sent by every station. The performance analysis of the two proposed protocols is introduced to evaluate their performance compared with Robust Super Poll protocol. The mathematical analysis and the experimental results show that the proposed protocols give higher throughput and lower overhead than Robust Super Poll protocol.

Keywords: Medium Access Control protocols, WLAN, Hub Polling Protocols, Robust Super Poll Protocol, Performance Analysis

1 Introduction

Wireless Local Area Network (WLAN) [1] is one of the most deployed wireless communication systems because of its simplicity, flexibility and low cost. It is used to extend the coverage area of the wired network. Currently, it is also used with the cellular networks, such as LTE network, to extend the service coverage area [2–4]. It can operate in two modes: the infrastructure mode and ad-hoc mode [5]. In the infrastructure mode, the point coordinator (PC) must be preinstalled and all the stations' transmission is relayed through it. However, in the ad-hoc mode, a group of stations can communicate together without the existence of PC.

Developing an efficient Medium Access Control (MAC) is essential to get the best performance of WLAN operations. The best performance is evaluated in terms of delays, collisions, channel utilization and energy consumption. The MACs for WLAN are classified into main three types [6–9]: contention methods, channelization methods and contention free methods.

In the contention methods, such as CSMA/CA, MACA, MACAW and IEEE 802.11 DCF, the stations content is on the channel resources. The station that gets the access to the channel sends data at the full rate of the channel. Collision occurs when several stations attempt to access the channel. In Carrier Sensing Multiple Access with Colision Avoidance (CSMA/CA) [10], the station, which has data to transmit, first checks whether the channel is busy or idle. If the channel is idle, the station starts transmitting its data frame. Otherwise, the station reschedules the transmission later. To eliminate collisions, CSMA/CA uses Binary Exponential Backoff (BEB). Based on BEB, the station chooses a random backoff time depending on the collision count. CSMA/CA can reduce the collisions using BEB, but it cannot prevent it. It also cannot handle the hidden and exposed terminal. It also causes a high delay in the high contention case.

Another contention protocol is called Multiple Access with Collision Avoidance (MACA) [11]. It is based on CSMA/CA and provides a method using the Request-To-Send (RTS) and Clear-To-Send (CTS) packets to overcome the hidden and exposed terminal problem. In MACA, sending station should send RTS packet to receiving station before sending a data frame. Upon receiving RTS packet, receiving station must send CTS packet to inform sending station and the others that it is ready to receive. MACA is preferred to CSMA/CA because it can reduce the collision using BEB, and it overcomes the hidden and exposed problem using RTS/CTS packets. However, it has some limitations: (1) sending station would increase the Backoff counter if

* Corresponding author e-mail: mahmoud.abdelsatar@fci.au.edu.eg

receiving station is busy and cannot send CTS packet; and (2) RTS/CTS presents more overhead, so it causes less throughput and utilization.

Multiple Access with Collision Avoidance for Wireless (MACAW) [12] is another contention protocol that is based on MACA. It sends Data Sending (DS) packet after RTS/CTS to inform the overhearing stations that the RTS/CTS exchange has been successful and data transmission is about to begin. In MACAW, if the receiving station is busy with another transmission and not ready to receive the data packet, it sends Request-For-Request-To-Send (RRTS) packet to the RTS sending station. The busy receiving station uses the RRTS packet to inform the sending station that it is busy and cannot receive data at the present. It can reduce collision using BEB. It overcomes the hidden and terminal problem as well as solves the problem of busy receiving station. Nevertheless, RTS/CTS, DS and RRTS messages present overheads.

IEEE 802.11 Distributed Coordination Function (DCF) method [13, 14] is proposed to utilize CSMA/CA, MACA or MACAW. In DCF, the sending station waits DCF Inter-Frame Space (DIFS) time, if the channel is found to be idle before starting transmission. Furthermore, the receiving station must wait Short Inter-Frame Space (SIFS) time before it responds back.

The second type of MACs for WLAN is channelization methods. This type includes Frequency Division Multiple Access (FDMA), Time Division Multiple Access (TDMA) and Code Division Multiple Access (CDMA) [15–18]. In this type, the channel data rate is shared among the stations. Each station sends data using its share of the full rate. No collisions can occur in the channelization method. In FDMA, the channel bandwidth is divided into a number of small frequency bands. Each frequency band is assigned to a station. In TDMA, the time is divided into frames. Each frame consists of a number of slots. Each slot is assigned to a station. The station can use only its slot to send data. In CDMA, a number of codes are used to support the multiple transmissions on the same frequency band. Each station uses a unique code. The codes are created in a way that avoids the collisions from different sources.

The contention free methods are a different type of MACs for WLAN. Polling methods are an example of this type in which each station has its turn to transmit data using the full rate of the channel.

Various papers are devoted to developing and optimizing the polling based medium access protocols [19, 20] because of the following: (1) the polling methods should provide bounded service delay with maximum and minimum access delays [7], (2) the data rates on the channel should be predicted and fixed [6], and (3) the polling is deterministic and suitable for the channels that are controlling some kinds of automated machines, such as Machine Type Communication (MTC) applications [6, 7]. The polling methods are classified into two types: Roll-call polling and hub polling. In the roll-call, the poling is fully centralized and the station cannot send its data frame until it receives a poll frame from the PC. In the hub polling method, the polling is partially distributed; a station can send its data frame directly after its previous station in the polling order finishes its transmission.

IEEE 802.11 standard Point Coordination Function (PCF) [19] is one of the protocols based on the polling method. It is a roll-call polling type where the PC is completely responsible for the polling operations. PC polls each station individually through sending a poll frame. Polling each station individually presents excessive overhead. Gans in [20] introduced the Robust Super Poll (RSPL) MAC to maximize the utilization of the channel. In RSPL, the PC polls multiple stations by broadcasting a single poll frame that contains the full polling list. In addition, it applies the chaining mechanism to ensure that every station receives the polling frame. The chaining mechanism means that each station appends the remaining of the polling list to its data frame. The chaining mechanism presents an overhead that is increased by the time due to the retransmission of the polling list after the transmission of the data.

The current paper presents two polling based MAC protocols. The first one is called Distributed List Hub Polling (DLHPL) protocol which is based on RSPL protocol. It is used for ad-hoc network that consists of a group of stations without PC. In DLHPL, the polling list is formed in a fully distributed manner instead of forming it centrally by PC in RSPL. It is likely fair in power consumption because all the stations contribute in forming the polling list. It also avoids the single point of failure. Moreover, the overhead reduces and throughput increases compared with RSPL because the station , in DLHPL, appends only the next station address instead of appending the remaining polling list in RSPL.

The second proposed protocol is called Light Robust Super Poll (LRSPL) protocol which is introduced to minimize RSPL's control overhead for the polling list formation and management. In LRSPL, only the polling list updates are broadcasted instead of broadcasting the full list in RSPL. To guarantee the delivery of the polling order to the stations, LRSPL uses the acknowledgment method. The PC sends order-update frame and waits an acknowledgement from the stations instead of the chaining mechanism, which is used by RSPL which introduces high overhead.

The present paper is outlined as follows. Section 2 describes RSPL MAC. Section 3 is devoted to the proposed DLHPL MAC. Section 4 addresses the proposed LRSPL MAC. Section 5 exhibits performance measures. Section 6 presents the experimental results.

2 Robust Super-Poll (RSPL) MAC Protocol

RSPL [20] is hub polling based MAC protocol. It is introduced to maximize the utilization of the channel.

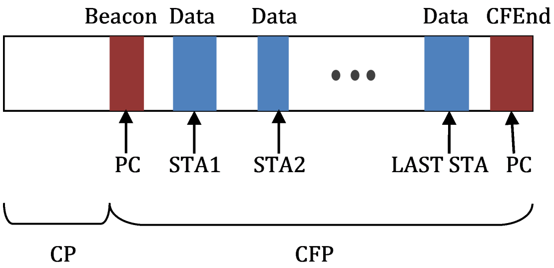

Fig. 1: RSPL super frame structure.

RSPL divides the time into frames called super frames; each frame consists of a Contention Free Period (CFP) and a Contention Period (CP) as shown in Fig. 1. In RSPL, the PC can poll more than one station in the same poll frame through broadcasting the full polling list instead of polling single station as in the standard PCF protocol. The polling list involves the addresses of the joined stations.

In the CP, the station that seeks to join the polling list uses the DCF method [14] with RTS/CTS technique to access the channel. CP is fixed length in time and starts after the PC broadcasts the CFEnd frame. At the end of the CP, the PC broadcasts the full polling list. In the CFP, each station in the polling list takes its turn to transmit its data followed by the part of the polling list that contains the next stations that have not transmitted yet.

The CFP length is variable depending on the number of stations listed in the polling list. CFP starts after sending the Beacon frame by the PC.

In RSPL, the polling list is formed and managed completely by the PC. The polling list is incrementally constructed; new joint stations are added to the current polling list. The station, which seeks to join the polling list, sends an association request to the PC. The PC responds to the association request with an Association ID (AID) which is used later as a key to address the devices in the polling list. At the end of the CP, the PC broadcasts a Beacon containing Contention Free Parameter Set (CFPS) elements as a sign of CFP starting. Upon receiving the Beacon frame, each station sets the Network Allocation Vector (NAV), which is used to ban a station from taking control of the medium during CFP. The PC follows the Beacon through broadcasting the polling list. The polling list consists of the AIDs of the stations. Upon receiving the polling list, if the station is the first in order, it starts transmission and appends the list of remaining stations. Otherwise, it recognizes its order from the polling list, the previous station and the maximum waiting time. The station cannot start transmission until it detects that its previous station finished transmission or the maximum waiting time elapsed. The PC broadcasts the CFEnd frame to inform the stations of the end of the CFP and to start the DCF mode. Upon receiving the CFEnd frame, the stations unset NAV and switch to DCF mode.

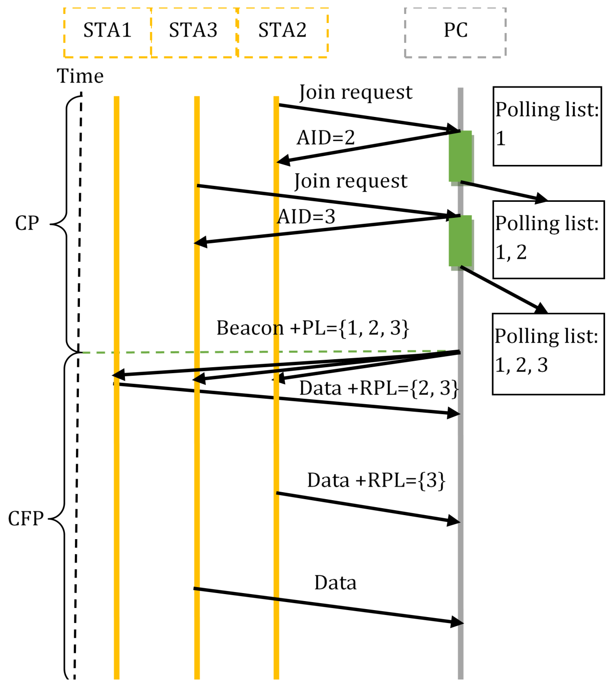

Fig. 2: Scenario of station joining example using RSPL.

Fig. 2 shows a scenario of stations that want to join the polling list. In the scenario, STA2 and STA3 want to join the Polling List (PL). In the beginning, the polling list contains only STA1 with AID equals to 1. When the PC receives the join requests from STA2 and STA3, the PC gives the AIDs of values 2 and 3 to STA2 and STA3 respectively and adds them to the polling list. After the end of the CP, the PC broadcasts the Beacon to inform the start of the CFP. Then, the PC broadcasts the full polling list. Upon receiving the full polling list, STA1 detects that it is the first. STA1 transmits its data frame followed by the remaining of the polling list. The Remaining Polling List (RPL) is the received polling list except STA1 AID. When STA2 detects that STA1 has finished the transmission, it transmits its data frame followed by the remaining polling list. Finally, STA3 transmits its data frame after finishing STA2 transmission.

Features of RSPL are defined as follows [20,21]: (1) It is partially distributed compared to PCF. In RSPL, the PC does not poll each station individually and the station can start transmitting as soon as its predecessor finishes, (2) it provides bounded delay service, and (3) the transmitting station transmits with the full rate. However, it has some drawbacks [20,21]: (1) The full polling list is broadcasted by the PC every CP even if update does not occur, (2) it guarantees the delivery of the polling list to each station by the chaining mechanism in which each device appends the remaining of the polling list after its data transmission,

(3) the exposure to the single point of failure using the PC, (4) PC must be supplied with high energy capacity to do its work, and (5) it is not fully distributed.

3 The Proposed Distributed List Hub Polling (DLHPL) MAC Protocol

The Proposed Distributed List Hub Polling (DLHPL) MAC is based on RSPL protocol and intends to overcome its drawbacks. It divides the time into frames called super frames, and each super frame consists of CP and CFP as shown in Fig. 3. In DLHPL, CSMA/CA is used as MAC protocol in the CP instead of the DCF method with RTS/CTS technique that is used in RSPL because CSMA/CA outperforms DCF in the case of hidden terminal absence based on Hung proof in [13].

In DLHPL protocol, the stations that seek to join the polling list use the CP to transmit the association request. This process is managed by a proposed distributed polling list formation method that will be discussed in section 3.1. The stations with critical data can transmit data in the CP. The CP is fixed length of time and it starts after broadcasting CFEnd frame by the last station in the polling list.

At the end of the CP, the stations use CFP to transmit its data. CFP is variable length of time based on the length of the polling list and it starts when the first station in the distributed polling list transmits a Beacon to inform all stations of the start of CFP. Then, the first station transmits its data frame including the next station address only instead of appending the remaining polling list. Upon receiving the Beacon, all other stations set the NAV. All stations hear the transmission of each other to check their transmission order. The station can start transmission when it finds that its address matches the next address that is appended to data frame of the previous station. Finally, the last station in the distributed polling list broadcasts the CFEnd to inform all stations of the end of CFP and the start of CP. The main procedure of DLHPL protocol will be described in section 3.3.

DLHPL has some advantages compared with other proposed algorithms: (1) It is fully distributed compared with RSPL. In DLHPL, both the polling list formation and the poll frames do not depend on a central point, such as the PC in RSPL. (2) It has less overhead than RSPL because the station, in DLHPL, receives its polling order as well as the previous and next stations addresses directly from another station in the group instead of chaining mechanism that is utilized in the RSPL in which all the remaining polling list is appended. (3) It is fair in power consumption because all stations contribute in the polling management. (4) It avoids the single point of failure because all stations share the role of PC. (5) It provides a bounded delay service. (6) The transmitting station transmits with the full rate. Despite the above-mentioned advantages, DLHPL does not handle the hidden terminal problem.

3.1 Polling List Formation

The polling list in DLHPL is built in a distributed manner between stations. It is arranged in ascending order based on priority value. The calculation of the priority value is described later in section 3.2. Each station maintains the values of its priority, Repeated Priority (RptdPriority) flag as well as the address and the priority of the previous station in the polling list, and the address of the next station in the polling list as shown in Fig. 4. It is possible that multiple stations have the same priority. The RptdPriority flag is used to handle the repeated priority value in the polling list formation. If multiple stations have the same priority value, the station with the RptdPriority value equals to 0 is responsible for responding to a station join request to guarantee that no more than one station responds at the same time and causes collision.

In DLHPL, the polling list formation is not centralized, i.e. it is distributed among the stations. The fully distributed polling list is identified by chaining the values that are distributedly stored in the stations. The station appends only the next station address to its data frame. It also does not append the remaining polling list like RSPL which reduces the overhead.

The polling list formation of DLHPL is different from the method that is used in the RSPL. In DLHPL, the

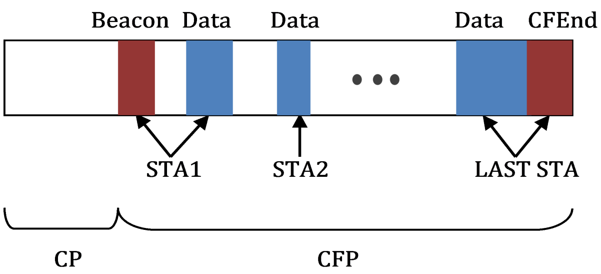

Fig. 3: DLHPL super frame structure.

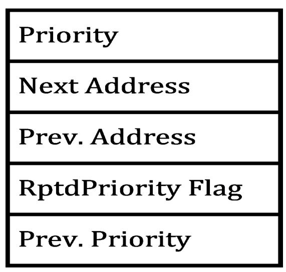

Fig. 4: Station's variables.

station that seeks to join the polling list broadcasts a join request frame involving its priority value. The structure of this frame is shown in Fig. 5. All the stations in the group receive the request and only one station, based on the attached priority value, can respond to this request to guarantee that only one transmission can occur. Assuming that the station with high priority value has lower priority, the station that responds to the request is the one that finds the attached priority value less than its priority and greater than its current previous station priority.

The station that accepts the join request edits the stored previous station address and priority, then it sends accept request frame that has structure shown in Fig. 5. The frame includes the address and priority values of the old previous station. The LST flag bit is set to 0 and the RptdPriority flag bit is set to 0. The LST flag bit is used to inform the new station if it is accepted as last one in the list. LST flag bit of 1 means that the station is accepted as last one. The new joined station changes its stored value of its next station address by the address of the station that accepted the join request. It also changes its stored value of its previous station address and priority by the values that are included in the accept request frame. The old previous station of the station that accepted the join request changes its stored next station address by the new joined station address.

There are special cases where the value of the priority is: (1) greater than the value of the last station, (2) less than the value of the first station, and (3) equals to another station priority value.

If the priority value is greater than the priority of the last station, the last station edits the stored next station address with the address of the new joined station. Then, it sends accept request frame with its address and priority values as previous address and previous priority. The LST flag bit is set to 1 and the RptdPriority flag bit is set to 0. On the other side, the new joined station sets the previous station address and priority values with the values of the last station that sends the accept request frame.

If the priority value is less than the priority of the first station, the first station edits the stored previous station address and priority with the new joined values. Then, it sends accept request frame with the LST flag bit which is set to 0 and the RptdPriority flag bit which is set to 0. The new joined station sets its next station address value with the address of the first station that sends the accept request.

If the priority value is equal to another station priority value, the station with RptdPriority flag value equals to 0 will respond to the join request. The station that accepts the join request edits the stored previous station address and priority. Then, it sends accept request frame with the address and priority values of the old previous station. The LST flag bit is set to 0 and the RptdPriority flag bit is set to 1.

A station is able to disjoin the distributed polling list. To disjoin the distributed polling list, the station broadcasts disjoin frame. The structure of this frame is shown in Fig. 6. The disjoin frame includes the previous address and priority as well as the next address as shown in the figure. When the disjoin frame is received by the stations, each of the next and previous stations updates its addresses and priorities. The next station edits its previous station address and priority to the attached previous address and priority. The previous station edits its next station address to the attached next address.

The joining and disjoining steps of the proposed protocol are defined as follows:

<u>Receive join request frame event in station *i:*</u>

1.RequestingStation_priority=
ExtractRequestingStationPriorityNumber(JoinRequestFrame)
2.RequestingStation_address=
ExtractRequestingStationAddress(JoinRequestFrame)

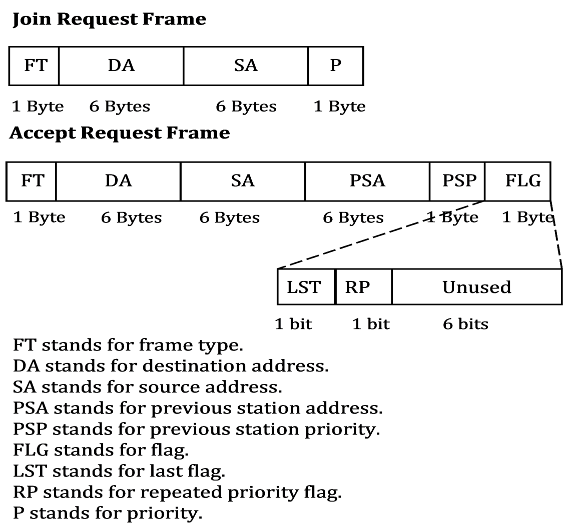

Fig. 5: Join Request and Accept Request frames.

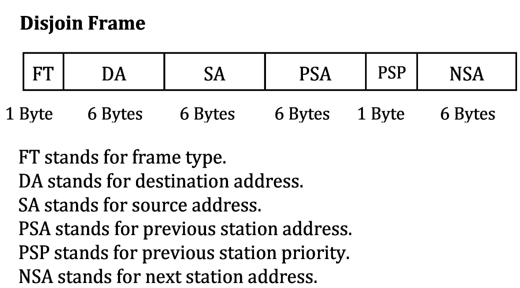

Fig. 6: Disjoin frame.

3.if((Station_priority$_i$ > RequestingStation_priority > PrevStation_Priority$_i$) or (PrevStation_address$_i$==NULL and Station_priority$_i$>RequestingStation_priority))
  (a)Station i sends AcceptRequest frame including the current PrevStation_address$_i$, PrevStation_priority$_i$ values as previous with the LST flag bit equals to 0 and RepeatedPriority flag bit equals to 0.
  (b)PrevStation_Priority$_i$= RequestingStation_priority
  (c)PrevStation_address$_i$= RequestingStation_address
4.else if(NextStation_address$_i$ == NULL and Station_priority$_i$ < RequestingStation_priority)
  (a)Station i sends AcceptRequest frame including the Station_address$_i$ and Station_priority$_i$ values as previous with the LST flag bit equals to 1 and RepeatedPriority flag bit equals to 0.
  (b)NextStation_Priority$_i$= RequestingStation_priority
  (c)NextStation_address$_i$= RequestingStation_address
5.else if(Station_priority$_i$ == RequestingStation_priority and RepeatedPriority$_i$ flag ==0)
  (a)Station i sends AcceptRequest frame involving the current PrevStation_address$_i$, PrevStation_priority$_i$ values as previous with the LST flag bit equals to 0 and RepeatedPriority flag bit equals to 1.
  (b)PrevStation_Priority$_i$= RequestingStation_priority
  (c)PrevStation_address$_i$= RequestingStation_address

<u>Receive Accept Request frame event in station *i:*</u>

1.RequestingStation_address= ExtractRequestingStationAddress(AcceptRequestFrame)
2.RespondingStation_address= ExtractRespondingStationAddress(AcceptRequestFrame)
3.RcvdPrevStation_address= ExtractPreviousStationAddress(AcceptRequestFrame)
4.RcvdPrevNode_priority= ExtractPreviousStationAddres(AcceptRequestFrame)
5.lastFlag= ExtractLastFlag(AcceptRequestFrame)
6.RepeatedPriorityFlag= ExtractRepeatedPriorityFlag(AcceptRequestFram)
7.if(Station_address$_i$ == RequestingStation_address)
  (a)if(lastFlag == 0)
    i.PrevStation_address$_i$= RcvdPrevStation_address
    ii.PrevStation_priority$_i$= RcvdPrevStation_priority
    iii.NextStation_address$_i$= ResponsingStation_address
  (b)else if(lastFlag ==1)
    i.PrevStation_address$_i$= RcvdPrevStation_address
    ii.PrevStation_priority$_i$= RcvdPrevStation_priority
    iii.NextStation_address$_i$= NULL
8.else if(Station_address$_i$ == RcvdPrevStation_address)
  (a)NextStation_address$_i$= RequestingStation_address

<u>Receive disjoin frame event in station i:</u>

1.RcvdNextStation_address= ExtractNextStationAddress(DisjoinFrame)
2.RcvdPrevStation_address= ExtractPreviousStationAddress(DisjoinFrame)
3.RcvdPrevStation_priority= ExtractPreviousStationPriority(DisjoinFrame)
4.if(Station_address$_i$== RcvdPrevStation_address)
  (a)NextStation_address$_i$= RcvdNextStation_address
5.else if(Station_address$_i$== RcvdNextStation_address)
  (a)PrevStation_address$_i$= RcvdPrevStation_address
  (b)PrevStation_priority$_i$= RcvdPrevStation_priority

Figs. 7-12, illustrate an example of 4 stations. STA4 sends a request to join the polling list. The example shows three scenarios when STA4 has priority 4, 1, and 8, respectively. Figs. 7 and 8, Figs. 9 and 10, and Figs. 11 and 12 show the stations variables values before STA4 sends the join request and the variables values after STA4 receives the accept request from STA3 in the three cases. In the first scenario, STA4 has a priority value equals to 4. STA4 broadcasts a join request with priority value of 4. All other stations receive the request and check the priority value. Based on the variables maintained in each station as shown Fig. 7, STA3 transmits an accept request frame and updates its previous address and priority to STA4 address and 4 as shown in Fig. 8. Upon receiving the accept request frame, STA1 and STA4 update the previous address and priority as well as the next address as shown in Fig. 8. STA 1 updates its next address to STA4 address instead of STA3 address. STA4 sets its previous priority and previous address to STA1 priority and address respectively, and it sets its next address to STA3 address. In the second scenario, STA4 has priority value equals to 1 that is less than that of the first station priority in the list. STA4 broadcasts a join request with priority value of 1. Based on the variables maintained in each station as shown Fig. 9, STA1 broadcasts an accept request frame and updates its previous address and previous priority to STA4 address and priority as indicated in Fig. 10. Upon receiving the accept request frame, STA4 sets its next address to STA1 address. In the third scenario STA4 has priority value equals to 8 that is greater than the last station priority value. STA4 transmits a join request frame with priority value of 8. Based on the variables that are maintained in each station as shown Fig. 11, STA2 broadcasts an accept request frame and updates its next address to STA4 address as shown in Fig. 12. Upon receiving the accept request frame, STA4 sets its previous address and priority to STA2 address and priority as shown in Fig. 12.

Accordingly, the list formation in DLHPL is better than RSPL because the list formation, in DLHPL, is fully distributed and does not depend on a single point.

3.2 Priority Value Selection

In DLHPL, the polling order of the station is defined based on the station priority value as discussed in section (3.1). Station priority is calculated using the Quality of Service (QoS) parameters for the application types presented in [22]. The station can select its priority value by combining the three QoS parameters: the real time, accuracy and the priority. DLHPL priority consists of 3 bits in which the most significant bit represents the

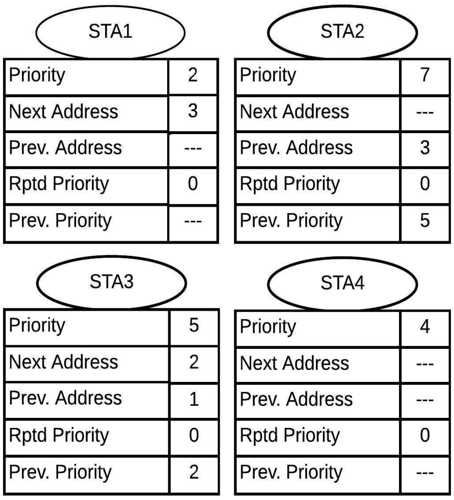

Fig. 7: Stations variables before STA4 sends join request with priority value of 4.

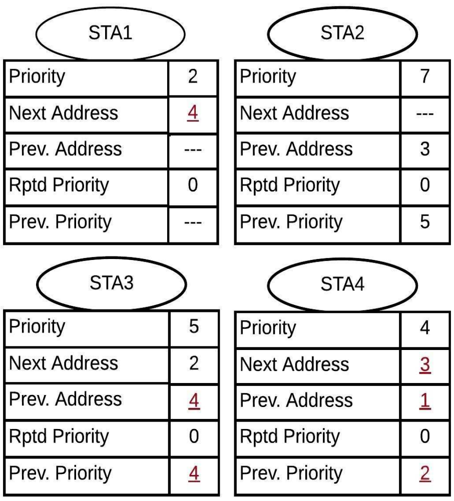

Fig. 8: Stations variables after accepting STA4 join request with priority value of 4.

real-time, the second bit represents accuracy, and the least significant bit represents priority. If any bit has value of 0, the station needs the service of real time, accuracy, and priority. The possible priority values for DLHPL stations are listed in Table 1.

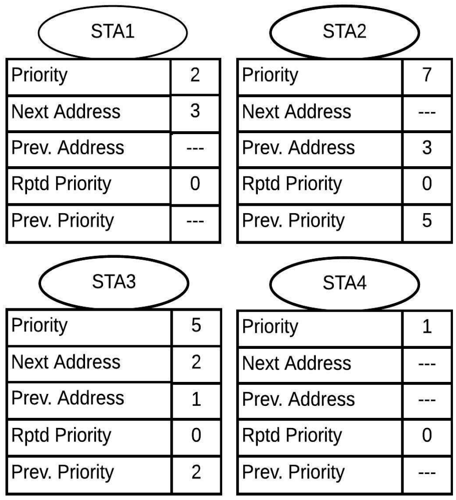

Fig. 9: Stations variables before STA4 sends join request with priority value of 1.

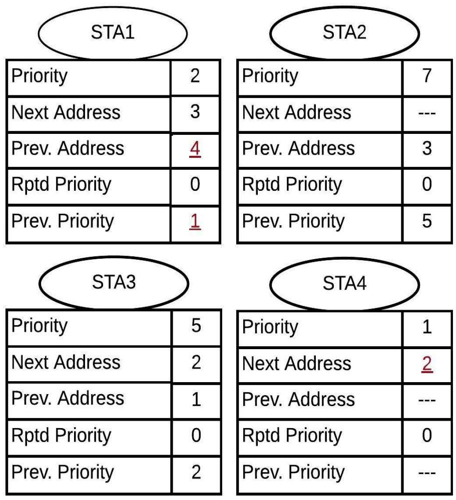

Fig. 10: Stations variables after accepting STA4 join request with priority value of 1.

3.3 DLHPL in Action

The overall steps of DLHPL procedure are divided into two phases: The joining phase and the transmission phase. The joining phase includes the steps involved in the joining operation of a new station. The transmission phase includes the steps involved in the transmission

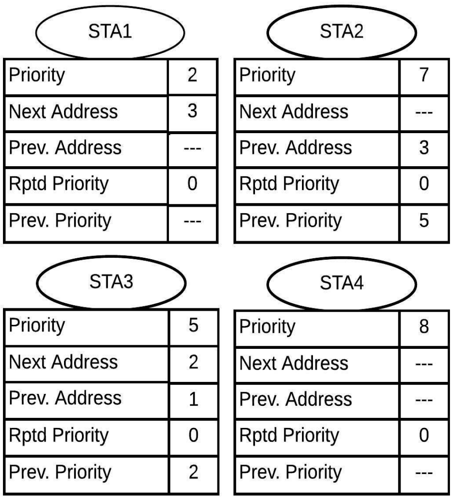

Fig. 11: Stations variables before STA4 sends join request with priority value of 8.

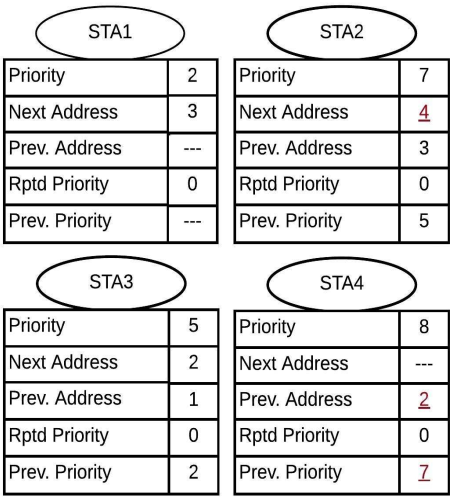

Fig. 12: Stations variables after accepting STA4 join request with priority value of 8.

operation for each station in the distributed polling list in order. The joining phase steps are defined, as follows:

1. Station i sets its priority value.
2. Station i detects the start of the CP when it receives the CFEnd frame that is broadcasted by the last station in the polling list.

Table 1: DLHPL possible priority values

Real-time	Accuracy	Priority	DLHPL Priority
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

3. Station i broadcasts a join request comprising its priority value.
4. A single station j responds by an accept request frame and edits the previous address and previous priority or the next. The accept request frame includes the previous station, P, address, the previous station priority and the next station, N, address.
5. Stations i, P, and N edit their previous address, previous priority, and next address.

The transmission phase steps are defined, as follows:

1. The first station in the polling order broadcasts the start of the CP and transmits its data frame or NTS frame appended by the next station address only.
2. Station i starts its transmission which recognizes its address as next station address after its previous station finishes the transmission.
3. The last station in the polling order transmits CFEnd frame to inform the others of the end of the CFP and the start of a new CP.

DLHPL is the first hub polling protocol that deploys the polling approach in a distributed manner on a group of stations without a PC. It has some advantages compared with other proposed algorithms. It is fully distributed compared with RSPL. It also has less overhead than RSPL because it does not use the chaining mechanism that appends the remaining polling list with the data. Moreover, it is fair in power consumption because all stations contribute in polling management. The delay is bounded, the station can transmit at the full rate, and it avoids the single point of failure.

4 The Proposed Light Robust Super-Poll (LRSPL) MAC Protocol

In this section, a simple modified version of the original RSPL called Light Robust Super Poll (LRSPL) is proposed. LRSPL seeks to reduce the overheads of RSPL, as follows: (1) PC transmits only the updates of the polling list individually for the updated stations instead of broadcasting the full polling list as in RSPL, (2) it does not use the chaining mechanism used by RSPL to ensure the delivery of the polling list to all the stations, but it uses the proposed method based on the acknowledgment mechanism.

In LRSPL, after the end of the CP, the PC sends a poll frame involving the previous station address and the order of the station that has a change in its order to every station that has an order update. The station order is used to calculate the maximum waiting time as mentioned in RSPL. The PC waits to receive an acknowledgment frame from the target station. If the PC did not receive an acknowledgement from a station, it would resend the polling frame again. After a fixed number of resending the polling frame without receiving an acknowledgement, the PC marks the station as out of reach and removes it from the polling list.

At the start of the CFP, the PC broadcasts a Beacon to inform the start of the CFP. Upon receiving the Beacon, all stations set the NAV flag and the first station in the polling order sends its data frame. In LRSPL, each station starts the transmission once it detects that its previous station finished its transmission. After the last station finishes its transmission, the PC broadcasts the CFEnd frame to inform the station of the end of the CFP and the start of the CP.

Fig. 13 shows a scenario of stations that seek to join the polling list. In the example, STA2 and STA3 seek to join the polling list. In the beginning, the polling list contains only STA1 with AID equals to 1. When the PC receives the join requests from STA2 and STA3, the PC gives two AIDs with values 2 and 3 to STA2 and STA3 respectively and adds them to the polling list. After the end of the CP, the PC sends poll frame to STA2 with previous station AID (PreAID) equals to STA1 AID and order equals to 2. Then, STA2 sends an acknowledgment to the PC. As soon as the PC receives an acknowledgment from STA2, it sends frame to STA3 with previous station AID equals to STA2 AID and order equals to 3. Then, the PC receives an acknowledgment from STA3.

Accordingly, LRSPL is more efficient than RSPL. In addition, it is reliable and has less overhead than RSPL because: (1) it uses the acknowledgment scheme instead of chaining scheme in RSPL, and (2) it sends only the updates of the list instead of sending the full list. However, it has some drawbacks: (1) it is not fully distributed, and (2) it uses PC which makes it vulnerable to a single point of failure.

4.1 LRSPL in Action

The overall steps of LRSPL procedure are divided into two phases: The joining phase and the transmission phase. The joining phase steps are defined, as follows:

1. Station i detects the start of the CP as soon as it receives the CFEnd frame broadcasted by the PC.
2. Station i sends a join request to the PC.
3. If the PC accepts the join request and replies to station i with an AID.

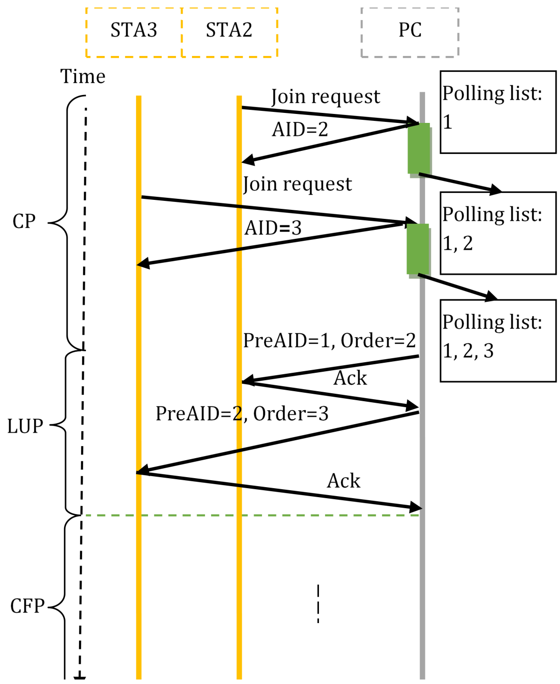

Fig. 13: Scenario of station joining example using LRSPL.

4. After the end of the CP, the PC sends the poll order individually, using the ARQ technique, to the new joined stations and to every station that has an order update.
5. Station i receives the poll order frame and recognizes its order and the previous station AID.

The transmission phase steps are defined as follows:

1. Once the PC finishes the transmission of the poll order frames, it broadcasts a Beacon to inform all stations of the CP start.
2. Upon receiving the Beacon, the first station in the polling order transmits its data frame or NTS frame.
3. Station i starts its transmission which recognizes its address as next station address after its previous station finishes the transmission.
4. After the transmission of the last station ends, the PC broadcasts the CFEnd frame.

5 Performance Measures

The unnormalized throughput, S, and the total control overhead, H, are used to evaluate the performance of the two proposed algorithms compared with RSPL MAC. S is defined as all data bytes sent successfully per time unit. It is the sum of the data bytes sent successfully in CP and CFP for each super frame along the simulation time. H is defined as the sum of all control bytes used in list formation and management of the polling operations, such as joining request frames, accept joining request frames, and poll frames.

S is obtained by:

$$S = \frac{N_{SF}(\overline{D_{CP}} + \overline{D_{CFP}})}{T_S} \tag{1}$$

Where $\overline{D_{CP}}$ and $\overline{D_{CFP}}$ are the average data bytes sent in CP and CFP respectively, T_S is the simulation time and N_{SF} is the total number of super frames. Assuming that no data exists be sent in CP, then $\overline{D_{CP}}$ equals Zero. In the following equations, it is assumed that any station sends one data frame in the CFP.

D_{CFP} is calculated using the following equation:

$$D_{CFP} = \sum_{i=1}^{N_D} L_{D_i} \tag{2}$$

Where N_D is the average number of stations that have data frames to be sent in a single CFP and L_{D_i} is the data frame length in bytes for the i^{th} station.

The probability generating function (PGF) of D_{CFP} is given by:

$$D_{CFP}(z) = E(z^{\sum_{i=1}^{N_D} L_{D_i}}) \tag{3}$$

L_{D_i} are independent and identically distributed random variables. Thus, equation (3) can be represented, as follows:

$$\begin{aligned} D_{CFP}(z) &= E(z^{L_{D_1}+L_{D_2}+\ldots+L_{D(N_D)}}) \\ &= E(z^{N_D L_D}) \\ &= E\left(\left(z^{L_D}\right)^{N_D}\right) \\ &= X(Y(z)) \end{aligned} \tag{4}$$

Where X(Z) and Y(Z) are the PGFs of N_D and L_D, respectively.

Considering that L_D follows the geometric distribution illustrated in the next equation:

$$f(L_D) = \propto (1-\propto)^{L_D} \tag{5}$$

So

$$Y(z) = \frac{\propto Z}{1-(1-\propto)Z} \tag{6}$$

Where $\propto$ is the probability of the frame end.

Substituting the PGF of L_D in equation (4) results in the following equation:

$$D_{CFP}(z) = X\left(\frac{\propto Z}{1-(1-\propto)Z}\right) \tag{7}$$

Considering that the number of the data frames, N_D, in particular CFP is given by binomial distribution, then the probability distribution of N_D, assuming that any station sends one data frame in the CFP, is given by the following equation:

$$f(N_D) = \binom{N}{i} P^{i} (1-P)^{(N-i)} \quad \text{for } 0 \leq i \leq N \tag{8}$$

Where N is the total number of stations and P is station probability for sending data frame in CFP. Using the binomial distribution for N_D as given in equation (8), $D_{CFP}(z)$ can be given by:

$$D_{CFP}(z) = \left((1-P) + P\left(\frac{\propto Z}{1-(1-\propto)Z}\right)\right)^N \tag{9}$$

The average length of D_{CFP} can be obtained by differentiating equation (9) and setting Z value to one.

$$\overline{D_{CFP}} = NP\left(\frac{1}{\propto}\right) \tag{10}$$

Substituting $\overline{D_{CP}}$ and $\overline{D_{CFP}}$ by their values in equation (1), S is given by:

$$S = \frac{N_{SF} NP(\frac{1}{\propto})}{T_s} \tag{11}$$

Equation (11) is used to obtain S value for the three MACs protocols: RSPL, LRSPL and DLHPL. All the terms of equation (11), except N_{SF}, are equal for the three MACs. The following subsections provide the customized S and H for the proposed LRSPL and DLHPL compared with RSPL.

5.1 RSPL Throughput and Control Overhead

In this section, the average throughput, S, and the average control overhead, H, for RSPL are obtained. To find throughput according to equation (11), the average number of super frames for RSPL, $N_{SF(R)}$, is needed to be obtained. It can be obtained by dividing the total simulation time by the average superframe length using the following equation:

$$N_{SF(R)} = \frac{T_S}{\overline{T_{CP(R)}} + \overline{T_{CFP(R)}}} \tag{12}$$

Where $\overline{T_{CP(R)}}$ is the average length of RSPL CP period in seconds and has a constant value, and $\overline{T_{CFP(R)}}$ is the average length of RSPL CFP period in seconds. $\overline{T_{CFP(R)}}$ can be calculated, as follows:

$$\overline{T_{CFP(R)}} = \frac{\overline{CFP_R}}{R} \tag{13}$$

Where $\overline{CFP_R}$ is the average length of RSPL CFP period in bytes and R is the channel bit rate in bytes.

RSPL CFP length, CFP_R, is the sum, in bytes, of the data frames, acknowledgments, NTS frames, CFEnd frame, and the total polling list overhead. CFP_R is given by:

$$\mathrm{CFP_R} = \sum_{i=1}^{N_D} (L_{D_i} + L_K) + (N - N_D)L_N + L_C + L_{POLLS} \quad (14)$$

Where L_K, L_N and L_C are the acknowledgment, NTS and CFEnd frames lengths, respectively. L_{POLLS} is the total polling overhead's bytes in a single CFP. It is the sum of all polling lists appended to the stations data frame and the main polling list that is broadcasted by the PC. The main polling length is NL_A. Any station, i, appends $(N-i)L_A$. L_A is the station's address length.

L_{POLLS} is given by:

$$L_{POLLS} = NL_A + \sum_{i=1}^{N-1} (N-i)L_A$$

then

$$L_{POLLS} = (N(N+1)L_A/\,2) \quad (15)$$

Substituting L_{POLLS} by its value in equation (14), $\mathrm{CFP_R}$ can be given by:

$$\mathrm{CFP_R} = \sum_{i=1}^{N_D} (L_{D_i} + L_K) + (N - N_D)L_N + L_C + (N(N+1)L_A/\,2) \quad (16)$$

The PGF of $\mathrm{CFP_R}$ is given by:

$$\mathrm{CFP_R}(z) = E\left(z^{\sum_{i=1}^{N_D} L_{D_i} - N_D(L_N - L_K)}\right) \times z^{(NL_N + L_C + (N(N+1)L_A/\,2))} \quad (17)$$

L_{Di} are independent and identically distributed random variables.

The first term of equation (17) can be represented as:

$$E\left(z^{\sum_{i=1}^{N_D} L_{D_i} - N_D(L_N - L_K)}\right) = z^{-N_D(L_N - L_K)} E\left(z^{L_{D_1} + L_{D_2} + \ldots + L_{D(ND)}}\right) = X(z^{-(L_N - L_K)} Y(z)) \quad (18)$$

Where X(z) and Y(z) are the PGFs of N_D and L_D, respectively. Thus, equation (17) can be viewed as:

$$\mathrm{CFP_R}(z) = X(z^{-(L_N - L_K)} Y(z)) \times z^{(NL_N + L_C + (N(N+1)L_A/\,2))} \quad (19)$$

L_D follows the geometric distribution, so Y(Z) can be expressed as given in equation (6).

Substituting the PGF of L_D by its value in equation (19) results in the following equation:

$$\mathrm{CFP_R}(z) = X\left(z^{-(L_N - L_K)}\left(\frac{\propto Z}{1-(1-\propto)Z}\right)\right) \times z^{(NL_N + L_C + (N(N+1)L_A/\,2))} \quad (20)$$

The number of the data frames, N_D, in particular CFP is assumed to have binomial distribution with the probability distribution function given in equation (8).

Using the binomial distribution of N_D, $\mathrm{CFP_R}(z)$ can be given by:

$$\mathrm{CFP_R}(z) = \left((1-P) + P\, z^{-(L_N - L_K)}\left(\frac{\propto Z}{1-(1-\propto)Z}\right)\right)^N \times z^{(NL_N + L_C + (N(N+1)L_A/\,2))} \quad (21)$$

The average length of $\mathrm{CFP_R}$ can be obtained by differentiating equation (21) and setting Z value to one.

$$\overline{\mathrm{CFP_R}} = N\left(P\left(\frac{1}{\propto} + L_K\right) + (1-P)L_N\right) + L_C + (N(N+1)L_A/\,2) \quad (22)$$

The average length of CFP in seconds, $\overline{T_{CFP(R)}}$, can be expressed using equation (13), as follows:

$$\overline{T_{CFP(R)}} = \frac{\overline{\mathrm{CFP_R}}}{R} = \frac{N\left(P\left(\frac{1}{\propto} + L_K\right) + (1-P)L_N\right) + L_C + (N(N+1)L_A/\,2)}{R} \quad (23)$$

Substituting $\overline{T_{CP(R)}}$ and $\overline{T_{CFP(R)}}$ by their values in equation (12), $N_{SF(R)}$ is given by:

$$N_{SF(R)} = \frac{RT_S}{R\overline{T_{CP(R)}} + N(P(\frac{1}{\propto} + L_K) + (1-P)L_N) + L_C + (N(N+1)L_A/2)} \quad (24)$$

Substituting $N_{SF(R)}$ by its value in equation (11), the throughput of RSPL, S_R, is given by:

$$S_R = \frac{RT_S PN(\frac{1}{\propto})}{R\overline{T_{CP(R)}} + N(P(\frac{1}{\propto} + L_K) + (1-P)L_N) + L_C + (N(N+1)L_A/2)} \quad (25)$$

In RSPL, the average total control, H_R, is the sum of CP average control overhead, $H_{CP(R)}$, and CFP average control overhead, $H_{CFP(R)}$. Then, H_R can be calculated, as follows:

$$H_R = H_{CP(R)} + H_{CFP(R)} \quad (26)$$

$H_{CP(R)}$ is the sum of the join requests and the accept requests frames lengths. $H_{CFP(R)}$ is the sum of the main polling list length broadcasted by the PC and the aggregate length of the polling lists that are appended to the stations data frames as well as the CFEnd frame length.

$H_{CP(R)}$ can be obtained by:

$$H_{CP(R)} = N_{SF(R)} N_A((1+K_R)L_J + L_R) \quad (27)$$

Where L_J and L_R are the join request and the accept join request average frames lengths, respectively, N_A is the average number of join request arrival per a single CP period, and K_R is the average number of join request retransmissions per request.

Substituting $N_{SF(R)}$ by its value from equation (24), $H_{CP(R)}$ is given by:

$$H_{CP(R)} = \frac{RT_S N_A((1+K_R)L_J+L_R)}{R\overline{T_{CP(R)}}+N(P(\frac{1}{\propto}+L_K)+(1-P)L_N)+L_C+(N(N+1)L_A/2)} \quad (28)$$

$H_{CFP(R)}$ is given by:

$$H_{CFP(R)} = N_{SF(R)}(L_{POLLS}+L_C) \quad (29)$$

Substituting $N_{SF(R)}$ and L_{POLLS} by their values from equations (24) and (15) respectively, $H_{CFP(R)}$ is given by:

$$H_{CFP(R)} = \frac{RT_S((N(N+1)L_A/2)+L_C)}{R\overline{T_{CP(R)}}+N(P(\frac{1}{\propto}+L_K)+(1-P)L_N)+L_C+(N(N+1)L_A/2)} \quad (30)$$

Substituting $H_{CP(R)}$ and $H_{CFP(R)}$ by their values from equations (28) and (30) respectively in equation (26), H_R is given by:

$$H_R = \frac{RT_S((N(N+1)L_A/2)+L_C+N_A((1+K_R)L_J+L_R))}{R\overline{T_{CP(R)}}+N(P(\frac{1}{\propto}+L_K)+(1-P)L_N)+L_C+(N(N+1)L_A/2)} \quad (31)$$

5.2 LRSPL Throughput and Control Overhead

As stated earlier, to find LRSPL average throughput and average control overhead, the average number of super frames for LRSPL, $N_{SF(L)}$, should be obtained. It is obtained using the following equation:

$$N_{SF(L)} = \frac{T_S}{\overline{T_{CP(L)}}+\overline{T_{CFP(L)}}+\overline{T_L}} \quad (32)$$

Where $\overline{T_L}$ is the sum of the times to send all the order-update frames and acknowledgments frames that are sent in List Update Period (LUP). In LUP, PC sends the order update frames to notify list updates after the CP period. $\overline{T_L}$ is given by:

$$\overline{T_L} = \frac{\overline{L_L}}{R} \quad (33)$$

Where $\overline{L_L}$ is the average of all order-update frames and acknowledgments frames lengths that are sent in LUP period. $\overline{L_L}$ is given by:

$$\overline{L_L} = N_A N_p((1+K_L+K_C)L_p+L_K) \quad (34)$$

Where N_p is the average number of order-update frame for a single join request, L_p is the order update-frame length that is used in LRSPL, as well as K_L and K_C are the average number of frame retransmissions due to loss and corruption, respectively.

Substituting $\overline{L_L}$ by its value in equation (33), $\overline{T_L}$ can be obtained by:

$$\overline{T_L} = \frac{N_A N_p((1+K_L+K_C)L_p+L_K)}{R} \quad (35)$$

LRSPL CFP length, CFP_L, is different from RSPL CFP_R. In LRSPL, each station appends only the next station address instead of the full list. Thus, CFP_L is given by:

$$CFP_L = \sum_{i=1}^{N_D}(L_{D_i}+L_K)+(N-N_D)L_N+L_C+NL_A \quad (36)$$

The PGF of CFP_L is given by:

$$CFP_L(z) = E\left(z^{\sum_{i=1}^{N_D} L_{D_i} - N_D(L_N-L_K)}\right) \times z^{(NL_N+L_C+NL_A)} \quad (37)$$

L_{Di} are independent and identically distributed random variables.

The first term of equation (37) can be represented as:

$$\begin{aligned} &E\left(z^{\sum_{i=1}^{N_D} L_{D_i} - N_D(L_N-L_K)}\right) \\ &= z^{-N_D(L_N-L_K)} E\left(z^{L_{D_1}+L_{D_2}+\ldots+L_{D(ND)}}\right) \\ &= X\left(z^{-(L_N-L_K)} Y(z)\right) \end{aligned} \quad (38)$$

Where $X(z)$ and $Y(z)$ are the PGFs of N_D and L_D, respectively. Thus, equation (37) can be represented as:

$$CFP_L(z) = X(z^{-(L_N-L_K)}Y(z)) \times z^{(NL_N+L_C+NL_A)} \quad (39)$$

Substituting the PGF of L_D from equation (6) in equation (39) results in the following equation:

$$\begin{aligned} CFP_L(z) &= X(z^{-(L_N-L_K)}(\frac{\propto Z}{1-(1-\propto)Z})) \\ &\times z^{(NL_N+L_C+NL_A)} \end{aligned} \quad (40)$$

The number of the data frames, N_D, in particular CFP is assumed to have binomial distribution with the probability distribution function given in equation (8).

Using the binomial distribution for N_D, $CFP_L(z)$ can be given by:

$$\begin{aligned} CFP_L(z) &= ((1-P)+Pz^{-(L_N-L_K)}(\frac{\propto Z}{1-(1-\propto)Z}))^N \\ &\times z^{(NL_N+L_C+NL_A)} \end{aligned} \quad (41)$$

The average length of CFP_L can be obtained by the following equation:

$$\overline{CFP_L} = N(P(\frac{1}{\propto}+L_K)+(1-P)L_N)+L_C+NL_A \quad (42)$$

$\overline{T_{CFP(L)}}$ is the average length of CFP in seconds. It can be expressed, as follows:

$$\overline{T_{CFP(L)}} = \frac{\overline{CFP_{(L)}}}{R}$$

$$=\frac{\mathrm{N}(\mathrm{P}(\frac{1}{\propto}+\mathrm{L_K})+(1-\mathrm{P})\,\mathrm{L_N})+\mathrm{L_C}+\mathrm{NL_A}}{\mathrm{R}} \quad (43)$$

$\overline{\mathrm{T_{CP(L)}}}$ is the average length of LRSPL CP period in seconds and it has a constant value.

Substituting $\overline{\mathrm{T_{CFP(L)}}}$ and $\overline{\mathrm{T_L}}$ by their values in equation (32), $N_{SF(L)}$ is given by:

$$\mathrm{N_{SF(L)}}=\frac{\mathrm{RT_S}}{\mathrm{R}\overline{\mathrm{T_{CP(L)}}}+\mathrm{N}(\mathrm{P}(\frac{1}{\propto}+\mathrm{L_K})+(1-\mathrm{P})\mathrm{L_N})+\mathrm{L_C}+\mathrm{NL_A}+\overline{\mathrm{L_L}}} \quad (44)$$

Substituting $\mathrm{N_{SF(L)}}$ by its value in equation (11), the throughput for LRSPL, S_L, is given by:

$$\mathrm{S_L}=\frac{\mathrm{RT_S PN}(\frac{1}{\propto})}{\mathrm{R}\overline{\mathrm{T_{CP(L)}}}+\mathrm{N}(\mathrm{P}(\frac{1}{\propto}+\mathrm{L_K})+(1-\mathrm{P})\mathrm{L_N})+\mathrm{L_C}+\mathrm{NL_A}+\overline{\mathrm{L_L}}} \quad (45)$$

In LRSPL, the total control, H_L, is given by:

$$\mathrm{H_L}=\mathrm{H_{CP(L)}}+\mathrm{H_{CFP(L)}}+\mathrm{H_{LUP(L)}} \quad (46)$$

In LRSPL, $H_{CP(L)}$ is the same as $H_{CP(R)}$ and is calculated from equation (28). $H_{CFP(L)}$ is the sum of the next address bytes that are appended to the stations data frames and the CFEnd frame length. $H_{CFP(L)}$ is given by:

$$\mathrm{H_{CFP(L)}}=\mathrm{N_{SF(L)}}(\mathrm{NL_A}+\mathrm{L_C}) \quad (47)$$

Substituting $\mathrm{N_{SF(L)}}$ by its value from equation (44), $H_{CFP(L)}$ is given by:

$$H_{\mathrm{CFP(L)}}=\frac{\mathrm{RT_S}(\mathrm{NL_A}+\mathrm{L_C}))}{\mathrm{R}\overline{\mathrm{T_{CP(L)}}}+\mathrm{N}(\mathrm{P}(\frac{1}{\propto}+\mathrm{L_K})+(1-\mathrm{P})\mathrm{L_N})+\mathrm{L_C}+\mathrm{NL_A}+\overline{\mathrm{L_L}}} \quad (48)$$

$H_{LUP(L)}$ is defined as the average control overhead in LUP period and can be calculated, as follows:

$$\mathrm{H_{LUP(L)}}=\mathrm{N_{SF(L)}}\overline{\mathrm{L_L}} \quad (49)$$

Substituting $\mathrm{N_{SF(L)}}$ by its value from equation (44), $H_{LUP(L)}$ is given by:

$$H_{LUP(L)}=\frac{\mathrm{RT_S}\overline{\mathrm{L_L}}}{\mathrm{R}\overline{T_{\mathrm{CP(L)}}}+\mathrm{N}(\mathrm{P}(\frac{1}{\propto}+\mathrm{L_K})+(1-\mathrm{P})\mathrm{L_N})+\mathrm{L_C}+\mathrm{NL_A}+\overline{\mathrm{L_L}}} \quad (50)$$

Substituting $H_{CP(L)}$, $H_{CFP(L)}$ and $H_{LUP(L)}$ by their values from equations (28), (48) and (50) in equation(46), H_L is given by:

$$\mathrm{H_L}=\frac{\mathrm{RT_S}(\mathrm{N_A}((1+\mathrm{K_R})\mathrm{L_J}+\mathrm{L_R})+(\mathrm{NL_A}+\mathrm{L_C})+\overline{\mathrm{L_L}})}{\mathrm{R}\overline{T_{\mathrm{CP(L)}}}+\mathrm{N}(\mathrm{P}(\frac{1}{\propto}+\mathrm{L_K})+(1-\mathrm{P})\mathrm{L_N})+\mathrm{L_C}+\mathrm{NL_A}+\overline{\mathrm{L_L}}} \quad (51)$$

5.3 DLHPL Throughput and Control Overhead

To calculate throughput in DLHPL protocol, equation (11) is used. The parameter $\mathrm{N_{SF(D)}}$ in equation (11) is calculated using the following equation:

$$\mathrm{N_{SF(D)}}=\frac{\mathrm{T_S}}{\overline{\mathrm{T_{CP(D)}}}+\overline{\mathrm{T_{CFP(D)}}}} \quad (52)$$

CFP length for DLHPL protocol, $\mathrm{CFP_D}$, is given by:

$$\mathrm{CFP_D}=\sum_{i=1}^{\mathrm{N_D}}(\mathrm{L_{D_i}}+\mathrm{L_K})+(\mathrm{N}-\mathrm{N_D})\mathrm{L_N}+\mathrm{L_C}+\mathrm{NL_A} \quad (53)$$

The PGF of $\mathrm{CFP_D}$ is given by:

$$\mathrm{CFP_D}(z)=E\left(z^{\sum_{i=1}^{\mathrm{N_D}}\mathrm{L_{D_i}}-\mathrm{N_D}(\mathrm{L_N}-\mathrm{L_K})}\right)\times z^{(\mathrm{NL_N}+\mathrm{L_C}+\mathrm{NL_A})} \quad (54)$$

The first term of equation (54) can be represented as:

$$\begin{aligned}&\mathrm{E}\left(z^{\sum_{i=1}^{\mathrm{N_D}}\mathrm{L_{D_i}}-\mathrm{N_D}(\mathrm{L_N}-\mathrm{L_K})}\right)\\&=z^{-\mathrm{N_D}(\mathrm{L_N}-\mathrm{L_K})}\mathrm{E}\left(z^{\mathrm{L_{D_1}}+\mathrm{L_{D_2}}+\ldots+\mathrm{L_{D(ND)}}}\right)\\&=\mathrm{X}\left(z^{-(\mathrm{L_N}-\mathrm{L_K})}\mathrm{Y}(z)\right)\end{aligned} \quad (55)$$

Where X(z) and Y(z) are the PGFs of $\mathrm{N_D}$ and L_D, respectively. Thus, equation (54) can be represented as:

$$\mathrm{CFP_D}(z)=\mathrm{X}(z^{-(\mathrm{L_N}-\mathrm{L_K})}\mathrm{Y}(z))\times z^{(\mathrm{NL_N}+\mathrm{L_C}+\mathrm{NL_A})} \quad (56)$$

Substituting the PGF of L_D by its value from equation (6) in equation (56) results in the following equation:

$$\mathrm{CFP_D}(z)=\mathrm{X}(z^{-(\mathrm{L_N}-\mathrm{L_K})}(\frac{\propto \mathrm{Z}}{1-(1-\propto)\mathrm{Z}}))\times z^{(\mathrm{NL_N}+\mathrm{L_C}+\mathrm{NL_A})} \quad (57)$$

Using the binomial distribution for N_D, $\mathrm{CFP_D}(z)$ can be given by:

$$\mathrm{CFP_D}(z)=\left((1-\mathrm{P})+\mathrm{P}z^{-(\mathrm{L_N}-\mathrm{L_K})}\left(\frac{\propto \mathrm{Z}}{1-(1-\propto)\mathrm{Z}}\right)\right)^{\mathrm{N}}\times z^{(\mathrm{NL_N}+\mathrm{L_C}+\mathrm{NL_A})} \quad (58)$$

$\overline{\mathrm{CFP_D}}$ can be obtained by the following equation:

$$\overline{\mathrm{CFP_D}}=\mathrm{N}(\mathrm{P}(\frac{1}{\propto}+\mathrm{L_K})+(1-\mathrm{P})\,\mathrm{L_N})+\mathrm{L_C}+\mathrm{NL_A} \quad (59)$$

$\overline{\mathrm{T_{CFP(D)}}}$ can be expressed, as follows:

$$\overline{\mathrm{T_{CFP(D)}}}=\frac{\overline{\mathrm{CFP_{(D)}}}}{\mathrm{R}}=\frac{\mathrm{N}(\mathrm{P}(\frac{1}{\propto}+\mathrm{L_K})+(1-\mathrm{P})\,\mathrm{L_N})+\mathrm{L_C}+\mathrm{NL_A}}{\mathrm{R}} \quad (60)$$

Substituting $\overline{\mathrm{T_{CFP(D)}}}$ by its value in equation (52), the average number of super frames for DLHPL, $N_{SF(D)}$, can be obtained by:

$$\mathrm{N_{SF(D)}}=\frac{\mathrm{RT_S}}{\mathrm{R\overline{T_{CP(D)}}}+\mathrm{N(P(\frac{1}{\propto}+L_K)+(1-P)L_N)+L_C+NL_A}} \quad (61)$$

Substituting $\mathrm{N_{SF(D)}}$ by its value in equation (11), the throughput of DLHPL, S_D, is given by:

$$\mathrm{S_D}=\frac{\mathrm{RT_SPN(\frac{1}{\propto})}}{\mathrm{R\overline{T_{CP(D)}}}+\mathrm{N(P(\frac{1}{\propto}+L_K)+(1-P)L_N)+L_C+NL_A}} \quad (62)$$

In DLHPL, the total control overhead, H_D, is given by the following equation:

$$\mathrm{H_D}=\mathrm{H_{CP(D)}}+\mathrm{H_{CFP(D)}} \quad (63)$$

In DLHPL, $H_{CP(D)}$ is the same as $H_{CP(R)}$ and it is calculated from equation (28). $H_{CFP(D)}$ is the same as $H_{CFP(L)}$ and it is calculated from equation (48).

Substituting $H_{CP(D)}$ and $H_{CFP(D)}$ by their values in equation (63), H_D is given by:

$$\mathrm{H_D}=\frac{\mathrm{RT_S(N_A((1+K_R)L_J+L_R)+(NL_A+L_C))}}{\mathrm{R\overline{T_{CP(D)}}}+\mathrm{N(P(\frac{1}{\propto}+L_K)+(1-P)L_N)+NL_A+L_C}} \quad (64)$$

6 Results

RSPL, LRSPL, and DLHPL MAC protocols are tested with designed simulator using the parameters listed in table 2. The results are obtained from the simulator according to the following assumptions: (1) No hidden terminal, i.e. all stations are in a single coverage area and can receive from each other, (2) all stations send data with an equal and fixed size, (3) the channel is reliable with no loss or interference, (5) no data frames are sent on the CP, (6) the propagation delay is neglected, (7) any station is permitted to send one packet in CFP at most. The simulation results are obtained more than once in different simulation times.

The protocols are tested using different maximum number of stations over fixed simulation time to investigate the effect of the number of stations on the performance.

Fig. 14 shows the overhead comparison between RSPL, DLHPL, and LRSPL with a different maximum number of stations over fixed simulation time. It exhibits that the control overhead of DLHPL and LRSPL is less than that of RSPL. The control overhead of RSPL increases with the increase in the number of stations compared with that of DLHPL and LRSPL. The simulation results are consistent with mathematical performance analysis. LRSPL overhead is lower than RSPL overhead because in LRSPL, PC broadcasts the polling list updates only after the joining success of one or more stations unlike RSPL in which the whole polling list is broadcasted periodically in every super frame. In addition, each station in LRSPL appends the next station address only after finishing its transmission unlike RSPL in which each station appends the remaining polling list which dramatically increases the overhead with the increase of the number of stations. The overhead of DLHPL is lower than that of RSPL because every station in DLHPL, which succeeds to join the polling list, receives its polling order as well as the previous and next stations addresses directly from another station in the group instead of broadcasting the whole polling list by the PC as in RSPL. Unlike the RSPL, no polling list exists in DLHPL to broadcast every super frame. As in LRSPL, every station, in DLHPL, appends only the next station address after finishing its transmission instead of the chaining mechanism that is utilized in the RSPL in which all the remaining polling list is appended. The results in Fig. 14 also show that the overhead of DLHPL is lower than that of LRSPL because in DLHPL when the station succeeds to join the polling list, the stations that need to update their information receive only a single message broadcasted from the station that accepts the join request instead of broadcasting a number of messages equal to the number of stations that need to update their information in LSRPL.

Fig. 15 shows the normalized throughputs of DLHPL, LRSPL, and RSPL with a different maximum number of stations over fixed simulation time. It exhibits that DLHPL and LRSPL throughput is higher than that of RSPL, because the number of super frames in DLHPL and LRSPL is higher than the number in RSPL due to the decrease of the super frame length in the proposed protocols. This reduction results from the decrease of the overhead in every super frame in DLHPL and LRSPL protocols.

Table 2: Simulation Parameters

Parameter	Value
Contention Period Time	3000 Microseconds
Arrivals rate per CP (Poisson)	1 station/CP
Max Back off	500 Microseconds
Data Rate	12 Mbps
DIFTime	50 Microseconds
SIFTime	50 Microseconds
Control Frame Length	20 Bytes
ACK Frame Length	20 Bytes
Data Frame Length	1024 Bytes
CFEnd Frame Length	20 Bytes

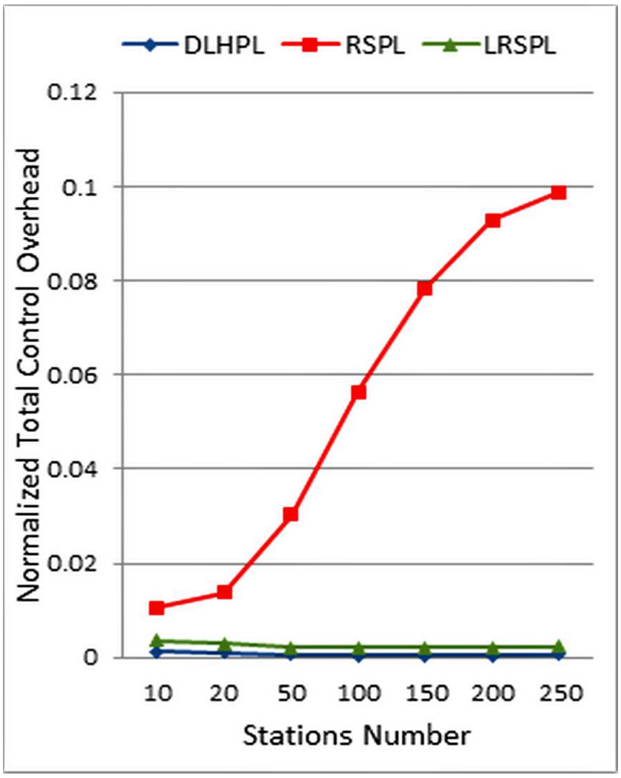

Fig. 14: Normalized overhead of RSPL, LRSPL and DLHPL for the simulations with a different maximum stations number.

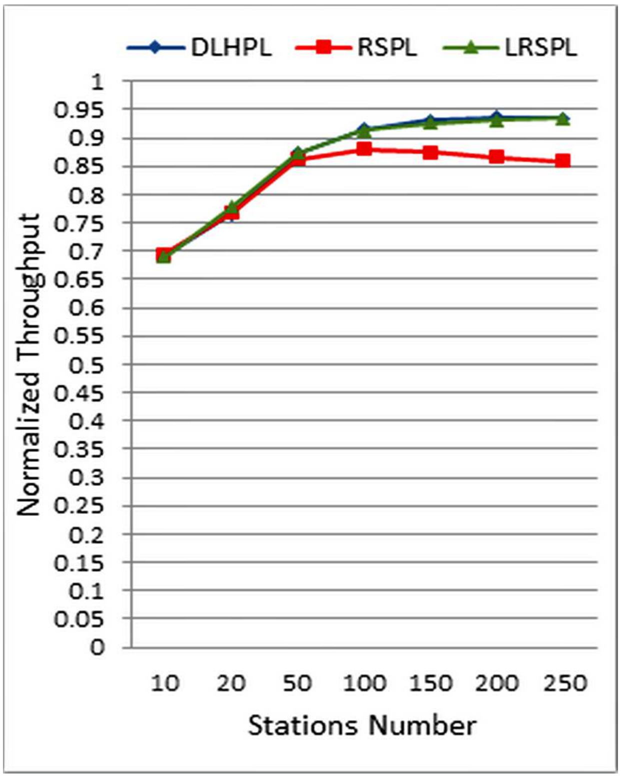

Fig. 15: Normalized throughput of RSPL, LRSPL and DLHPL for the simulations with a different maximum stations number.

7 Conclusion

In this paper, two WLAN hub polling MAC protocols (i.e. Distributed List Hub Polling (DLHPL) and Light Robust Super-Poll (LRSPL)) were proposed to enhance the performance of Robust Super-Poll (RSPL) protocol. Throughput and overhead were used as performance measures. Both proposed protocols reduced the overhead by eliminating the use of the chaining mechanism that is used in RSPL in which all the remaining polling list is appended to every data frame that is sent by every station. Every station in DLHPL and LRSPL appends only the next station address after finishing its transmission. Unlike the RSPL, in DLHPL and LRSPL, the polling list is not broadcasted every super frame. DLHPL utilizes the hub polling approach without using a PC. It equivalently distributes the polling management operations among the stations. The polling list in DLHPL is distributedly and dynamically formed. However, LRSPL still uses PC, but it enhances the reliability through the acknowledgment approach instead of using RSPL's chaining mechanism. In LRSPL, the PC broadcasts polling list updates based on the joining event only instead of broadcasting the whole polling list periodically in every super frame in RSPL. DLHPL overhead is lower than that of LRSPL because in DLHPL, the station that accepts the join request sends only a single broadcasted message to all the stations that need to be updated. However, in LSRPL, the PC broadcasts a number of messages equal to the number of the updated stations. Decreasing the overhead in every super frame in the two proposed protocols reduced the length of the superframe which increased the number of super frames compared with RSPL. Increasing the number of super frames resulted in transmitting more data bytes in the unit time. Accordingly, the throughput of the two proposed protocols increased compared with RSPL. The mathematical performance analysis and the simulation results proved that DLHPL and LRSPL provide higher throughput and lower overhead than RSPL.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this article

References

[1] B. Bellalta, L. Bononi, R. Bruno, and A. Kassler, Next generation IEEE 802.11 Wireless Local Area Networks:

Current status, future directions and open challenges, Computer Communications, 75, 1-25 (2016).
[2] R. M. Huq, K. P. Moreno, H. Zhu, J. Zhang, O. Ohlsson, and M. I. Hossain. On the Benefits Of Clustered Capillary Networks For Congestion Control In Machine Type Communications Over LTE, in Proc. International Conference on Computer Communication and Networks (ICCCN), 1-7, (2015).
[3] A. Azari, and G. Miao. Energy efficient MAC for cellular-based M2M communications, in Proc. 2nd IEEE Global Conference on Signal and Information Processing, 128-132, (2014).
[4] G. Rigazzi, N. K. Pratas, P. Popovski, and R. Fantacci. Aggregation and trunking of M2M traffic via D2D connections, in Proc. IEEE International Conference on Communications (ICC), 2973-2978, (2015).
[5] M. G. Rubinstein, I. M. Moraes, M. E. M. Campista, L. H. M. Costa, and O. C. M. Duarte. A Survey on Wireless Ad Hoc Networks, in Proc. IFIP International Conference on Mobile and Wireless Communication Networks, 1-33, (2006).
[6] S. S. Rao, S. V. Chalam, and D. S. Rao, A Survey On Mac Protocols For Wireless Multimedia Networks, International Journal of Computer Science and Engineering Survey (IJCSES), 2, 5-74 (2011).
[7] Q. Ni, L. Romdhani, and T. Turletti, A survey of QoS enhancements for IEEE 802.11 wireless LAN, Wireless Communications and Mobile Computing, 4, 547-566 (2004).
[8] A. Malika, J. Qadir, B. Ahmad, K. A. Yau, and U. Ullah, QoS in IEEE 802.11-based wireless networks: A contemporary review, Network and Computer Applications, 55, 24-46 (2015).
[9] H. Al-Mefleh and O. Al-Kofahi, Frequency-domain contention and polling MAC protocols in IEEE 802.11 wireless networks: A survey, Computer Communications, 129, 1-18 (2018).
[10] E. Ziouva and T. Antonakopoulos, CSMA/CA performance under high traffic conditions: throughput and delay analysis, Computer Communications, 25, 313-321 (2002).
[11] P. Karn. MACA-a new channel access method for packet radio, In Proc. ARRL/CRRL Amateur radio 9th computer networking conference, 134-140, (1990).
[12] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, MACAW: a media access protocol for wireless LAN's, ACM SIGCOMM Computer Communication Review Newsletter, 24, 212-225 (1994).
[13] F. Y. Hung and I. Marsic, Performance analysis of the IEEE 802.11 DCF in the presence of the hidden stations, Computer Networks, 54, 2674-2687 (2010).
[14] G. Bianchi, Performance analysis of the IEEE 802.11 distributed coordination function, IEEE Journal on Selected Areas in Communications, 18, 535-547 (2000).
[15] T. E. Gorsuch and C. Amalfitano. Dynamic bandwidth allocation to transmit a wireless protocol across a code division multiple access (CDMA) radio link, U.S. Patent 6 081 536, June 27, (2000).
[16] Y. P. Fallah, S. Khan, P. Nasiopoulos, and H. Alnuweiri. Hybrid OFDMA/CSMA based medium access control for next-generation wireless LANs, in Proc. IEEE International Conference on Communications, 2762-2768, (2008).
[17] Z. J. Haas and D. A. Dyson. The dynamic packet reservation multiple access scheme for multimedia traffic, in Proc. IEEE International Conference on Communications, 1640-1644, (1998).
[18] T. S. Rappaport, Wireless communications: principles and practice, Prentice Hall PTR, New Jersy USA, 395-436, (1996).
[19] S. A. Rasheed, K. Masnoon, N. Thanthry, and R. Pendse. PCF vs DCF: A Performance Comparison, in Proc. Thirty-Sixth Southeastern Symposium in System Theory, 215-219, (2004).
[20] A. Ganz, A. Phonphoem, and Z. Ganz, Robust Superpoll with Chaining Protocol for IEEE 802.11 Wireless LANs in Support of Multimedia Applications, Wireless Networks, 7, 65-73 (2001).
[21] Z. Liqiang, Z. Jie, and Z. Hailin, Hub polling based IEEE 802.11 PCF with integrated QoS differentiation, Wireless Communications and Mobile Computing, 9, 1220-1230 (2009).
[22] R. Liu, W. Wu, H. Zhu, and D. Yang. M2M-oriented QoS categorization in cellular network, in Proc. 7th International Conference on Wireless Communications, Networking and Mobile Computing, 1-5, (2011).

Mahmoud Abdelsattar Mohammed received the B.Sc degree in Information Technology from the Faculty of Computers and Information, Assiut University, Assuit, Egypt, in 2012. He is working as Teaching Assistant at the Information Technology Department, Faculty of Computers and Information, Assiut University, Assiut, Egypt, from 2012 to now.

Nagwa M. Omar received the B.Sc., M.Sc., and PhD degrees in Computer Engineering from the Faculty of Engineering, Assiut University, Assiut, Egypt, in 1999, 2002, and 2008 respectively. She worked as Assistant Professor at the Information Technology Department, Faculty of Computers and Information, Assiut University, Assiut, Egypt, from 2009 to 2016. She is working as Associate Professor at the Information Technology Department, Faculty of Computers and Information, Assiut University, Assiut, Egypt from 2016 to now.

Hosny M. Ibrahim received the B.Sc., and M.Sc. degrees in Electrical Engineering from the Faculty of Engineering, Assiut University, Assiut, Egypt, in 1973, and 1977 respectively. He received the Ph.D. degree in Electrical Engineering from Iowa State University, Ames, Iowa, U.S.A. in 1982. He was the Dean of the Faculty of Computers and Information, Assiut University, Assiut, Egypt from September 2002 to August 2011. He was the head of the Information Technology Department, Faculty of Computers and Information, Assiut University, Assiut, Egypt from July 2010 to May 2015. He is currently Professor at the Information Technology Department, Faculty of Computers and Information, Assiut University, Assiut, Egypt.