

# Adaptive device-to-device communication using Wi-Fi Direct in smart cities

Osama M. F. Abu-Sharkh<sup>1</sup> · Esam AlQaralleh<sup>1</sup> · Omar M. Hasan<sup>2</sup>

Published online: 4 May 2016  
© Springer Science+Business Media New York 2016

**Abstract** Wi-Fi Direct has become vastly popular in the last few years. Due to its fast network setup and the provision of high transmission rates, it is expected to be the piggyback technology for high speed device-to-device communications in smart cities. The increasing demand on Wi-Fi Direct networks and its co-location in the same band with other Wi-Fi ad-hoc and infrastructure networks have entailed the need for developing new adaptive techniques to utilize the wireless spectrum efficiently. In this paper, we introduce, analyze and implement a self-organizing algorithm designed specifically for pop-up Wi-Fi Direct networks. The algorithm dynamically changes the operating channels of pop-up Wi-Fi Direct networks according to their performance in a non-cooperative manner. It utilizes the spectrum efficiently, reduces congestion and enhances performance. Moreover, the modified Wi-Fi Direct network running the proposed algorithm is interoperable with all other Wi-Fi network modes and setups and does not affect their functionalities. On the contrary, it enhances their performance implicitly. We implement the algorithm using the driver of RTL8188CUS chipset in a Linux environment and conduct experiments to evaluate its performance. The obtained results illustrate the benefits of using the proposed algorithm.

**Keywords** Device-to-device · IEEE802.11 · Smart cities · Wi-Fi Direct

## 1 Introduction and motivation

Smart cities are information-intensive cities wherein the significance of networking and communication technologies is rising. Wireless networks have been, and still, vividly deployed in the last two decades with the aid of the recent advances and contiguous development of several wireless technologies and standards. Though, with the forthcoming massive deployment of wireless networked devices in smart cities, unprecedentedly, a large amount of data shall be produced and transferred over the air which urges the necessity to further enhance the current wireless technologies and make them meet the requirements and the increasing needs of the wide-ranging applications of smart cities. Smart, mobile and handheld devices such as smartphones and tablets are among the most usable networked devices in smart cities. Due to their high mobility, such devices get in close proximity of each other frequently. When a data transfer is desired between two of these networked devices, employing a device-to-device communication by making them communicate directly and taking other devices, such as base stations/access points, out of the data communication loop, enhances the performance of the designated networks and makes a more resourceful utilization of the scarce wireless spectrum.

Wi-Fi [1] has achieved a huge success in serving many networking applications efficiently over the last decade and a half. Wi-Fi Direct [2] has emerged recently, as one of the modes of Wi-Fi, to strengthen the use of direct device-to-device communication in wireless local area networks (WLANs) and overcome the drawbacks of the legacy ad-

---

✉ Osama M. F. Abu-Sharkh  
osama@psut.edu.jo

<sup>1</sup> Department of Computer Engineering, King Abdullah II Faculty of Engineering, Princess Sumaya University for Technology, Khalil Alsaket Street, Al-Jubaiha, P. O. Box 1438, Amman 11941, Jordan

<sup>2</sup> Department of Communications Engineering, King Abdullah II Faculty of Engineering, Princess Sumaya University for Technology, Khalil Alsaket Street, Al-Jubaiha, P. O. Box 1438, Amman 11941, Jordan

hoc mode. Wi-Fi Direct reduces the setup overhead and makes a fast and direct connection between two wireless devices or more without the involvement of any infrastructural setup. Yet, at least one of the involved wireless stations of a Wi-Fi Direct network shall have some of the access point capabilities. Wi-Fi Direct network is established on demand and for short periods of time in most of its used applications and hence it can be considered as a pop-up network.

Wi-Fi has recently gained the insight of many standardization bodies of other wireless technologies in the licensed and unlicensed bands. In the unlicensed band, Bluetooth [3] has been the de facto standard for wireless device-to-device communications for the last decade and a half. It has been essentially introduced for this purpose and it suits low power devices. Unfortunately, the data rate that can be achieved is very limited and cannot meet the demand of current applications. Therefore, the IEEE802.15 working group introduced in Ver.3.0 and above, the ability to use an alternative MAC and PHY such as Wi-Fi Direct for conveying Bluetooth profile data when high data rates are needed. Current research fingers are also pointing toward Wi-Fi to carry some of the traffic of cellular networks to reduce congestion and utilize the spectrum proficiently. In cellular networks, device-to-device communication can be distinguished in different approaches. One possible approach is the inclusion of the cellular network core or the associated base station to control the direct link communication between the involved devices. Long term evolution advanced (LTE-A) [4] may utilize this approach in the licensed and/or unlicensed bands to enhance the performance and increase capacity. Another possible approach is completely conveying the direct device-to-device transmission to another free wireless technology that meets the high speed demand of recent applications such as Wi-Fi Direct. The latter approach can be utilized when the communication occurs between two mobile devices which are in good proximity of each other since the range of Wi-Fi is less than the ranges of cellular technologies. Which approach is more efficient than the other is still vague and extensive research has been conducted recently adopting either approach [5–12]. One of the advantages of Wi-Fi over other wireless technologies including LTE-A is its high achievable data rates. Wi-Fi data rates currently reach hundreds of megabit per second with the wide use of IEEE802.11n [13], hit few gigabit per second with the newly standardized IEEE802.11ac [14] and reach almost 7Gbps with the use of IEEE802.11ad (WiGig) [15]. The aforementioned standards are already finalized and their products are currently available in the market. Furthermore, the data rate of Wi-Fi is expected to exceed 10 Gbps using IEEE802.11ax [16] upon its completion. On the other hand, enhancing the data rate of LTE-

A is still in progress and the goal is to reach 1.5Gbps in the uplink and few gigabit per second in the downlink [4]. The speed of a direct link between two devices in LTE-A is still unclear. Wi-Fi Direct can use any of the current and possible future amendments of the IEEE802.11 standard and hence can utilize the provided high speeds. Hence, Wi-Fi Direct fits well in concept to be the de facto technology for *fast* device-to-device communication in heterogeneous networks. Nevertheless, the current wireless free bands; 2.4 and 5 GHz used by most of the IEEE802.11 devices, are heavily loaded and barely serve the current needs of the existing Wi-Fi networks since the numbers of defined channels in these bands are limited. In the 2.4 GHz, Industrial Scientific Medical (ISM) band, there are only three to four non-overlapping channels while there are around twenty-three channels in the 5 GHz band in most countries. The default bandwidth of each channel of either band is 20 MHz. In IEEE802.11ac, many 20 MHz channels can be aggregated together to form one channel with a total bandwidth of up to 160 MHz. This shall reduce the number of available channels in the 5 GHz band dramatically. Therefore, even with the twenty-three 20 MHz channels in the 5 GHz band, the congestion is still considerable. Moreover, all wireless stations deploying the IEEE802.11 standard; whether they are associated to the same network or different networks and whether they use infrastructure, ad-hoc or Wi-Fi Direct modes, compete equally in accessing the channels. Hence, the opportunity to access a channel by any station is less likely to happen with the increasing number of competing stations. Consequently, a decrease in the achieved throughput, an increase in the average MAC delay and finally an increase in the collision rate shall occur. Handing off traffic from other wireless technologies shall make the congestion in these bands even worse.

The limited number of available channels in the 2.4 and 5 GHz frequency bands, the increasing use of Wi-Fi infrastructure and ad-hoc networks in these bands and the dynamic occurrence of pop-up Wi-Fi Direct networks make the currently defined fixed channel allocation strategy unfeasible in meeting the current and future demands of smart cities' applications.

Recently, High Efficiency WLAN Study Group has been formed by IEEE with a task to enhance system throughput/area by improving the spectrum efficiency in dense deployments [16–18]. The project is still in its early stages with no issued drafts and expected to be completed in 2019.

In this paper, we focus on evaluating and enhancing the performance of Wi-Fi Direct to reveal its efficiency in handling traffic in dense environment and consequently reflects its capability of serving the many applications which require device-to-device communication in smart

cities. We introduce a self-organizing algorithm designed specifically for pop-up Wi-Fi Direct networks in dense deployments to reduce congestion and utilize the scarce spectrum efficiently. The main contributions of our work and the differences from others in the literature are explained thoroughly in the following section.

The rest of the paper is organized as follows. We survey the related work in the literature and provide our contribution in Sect. 2. In Sect. 3, we introduce and discuss the design of the algorithm. We then discuss the experimental and simulation setups and provide evaluations and comparisons in Sect. 4. Finally, we conclude our paper in Sect. 5.

## 2 Literature review and contribution

Many studies have been conducted recently in the literature that address the reduction of congestion that may happen because of the increasing number of wireless networks that compete on the very limited resources in the unlicensed bands. All of them rely on sharing the spectrum wisely based on a real time analysis. We address some of the works that consider the dense deployments of IEEE802.11 wireless networks in the following.

In [19], Zhang et al. studied throughput maximization in multi-cell WLANs with access points. They optimized network performance based on joining channel assignment, association control, airtime sharing and contention window tuning. Kauffmann et al. introduced algorithms in [20] that minimize interference between competing access points and fairly associate different users to the access points. Xu et al. presented in [21] a game theoretical approach for channel assignment and user association in IEEE802.11 infrastructure wireless networks. They considered channel assignment and users association for balancing traffic load of access points operating on different channels as a non-cooperative game. In [22], Wang et al. studied the joint optimization of the following tasks, client association to select a base station to associate with, antenna beam selection when directional antennas are used, link scheduling to ensure conflict freedom and power adaptation to reduce mutual interference. They proposed a unified conflict-free scheduling algorithm that solves the joint optimization problem. In [23], Karimi et al. examined the proportional fair access point association with densely deployed wireless networks. They enabled collaboration and sharing among individual networks and presented optimized solutions for multiple access points to collaboratively serve wireless stations within a set of networks that share the same upstream provider. In [24], Ozyagci investigated the impact of both the association of wireless stations with access points and the positions of the access points on the aggregate throughput in a densely deployed

wireless LANs. They found that the impact of the former is more considerable than the latter. In [25, 26], Gong and Yang introduced distributed channel assignment algorithms for IEEE802.11n WLANs with heterogeneous clients. They concentrated in their work on distributing channels between infrastructure WLANs taking into consideration the enhanced MAC techniques introduced in the IEEE802.11n amendment of the standard. On the other hand, Ji et al. considered in [27] the enhanced MAC and PHY techniques of IEEE802.11ac amendment of the standard. They introduced a new MAC protocol specifically designed for IEEE802.11ac that reduces congestion and satisfies the quality of service of overlapping networks. In [28], Yue et al. introduced a distributed channel assignment algorithm for uncoordinated access points with their associated wireless stations where access points can configure their operating channels depending on their local traffic information. In [29], Chaves et al. proposed a mechanism that adaptively adjusts transmitting powers of wireless devices to improve Wi-Fi spatial capacity. In [30], Jiang et al. introduced a heuristic algorithm where two-dimensional optimization integrating channel assignment and power control are adopted. They considered both download and uplink traffic in their analysis. In [31], Jamil et al. claimed that transmitting power adjustment is not always possible to implement due to hardware and licensing limitations and it behaves aggressively towards transmitters having lower transmit power. They suggested the use of the clear channel assessment (CCA) threshold adaptation instead. In [32], a dynamic frequency allocation scheme was proposed that considers the absolute number of collided packets as a performance metric to move between channels in the 2.4 GHz frequency band. The well-known Wi-Fi company, Aruba Networks, developed a patented technique denoted by Adaptive Radio Management [33] that uses automatic infrastructure-based controls to manage channels.

All of the aforementioned works considered infrastructure mode of wireless LANs where standalone and/or controlled access points are involved and many wireless stations are associated with them.

In a recent work, Sagari et al. proposed in [34] an adaptive channel assignment scheme to enhance performance of mobile hotspots in dense environment. They assumed in their work that the data rate of the established hotspot is limited to a median uplink and downlink throughput up to 6 and 13 Mbps respectively which is not true in most cases. They also took into consideration, in designing the scheme, only one mobile hotspot among many fixed access points. Hence, they did not consider the congestion that may also occur between the mobile hotspots themselves in a dense deployment.

Rate adaption algorithms were proposed in many works in the literature to improve the network performance in

dense deployments. Some of the recent algorithms can be found in [35–37]. Rate adaptation algorithms are needed when the channel operating conditions are the main source of transmission errors and losses rather than the congestion. Therefore, the introduced algorithms solve mainly the problem of mistakenly lowering the transmission rates when the cause of the packet loss is the congestion. They do not actually solve the congestion problem itself. It is worth mentioning that transmitting at low data rates leads to a dramatic degradation in the overall performance according to the well-known performance anomaly problem which was firstly introduced in [38].

In [39], Abinader et al. evaluated three MAC access mechanisms of the IEEE802.11 standard, Distributed Coordination Function (DCF), Power-Save Multi-Poll (PSMP) and Hybrid Coordination Function (HCF). They concluded in their paper that in spite of the fact that the former is the worst in terms of performance, the other two mechanisms also suffer from a decrease in performance in dense deployments.

Our work differs from all of the above mentioned efforts and from others in the literature by adopting the following aspects in our design. In spite of the many works in the literature that consider WLANs in dense deployments [19–39], to the best of our knowledge, we are the first who introduce an adaptive performance-driven algorithm designed specifically for pop-up Wi-Fi Direct networks and not for the other modes and setups of the IEEE802.11 standard. The algorithm dynamically interchanges the operating channel based on the experienced performance at the MAC level. We target in our work pop-up Wi-Fi Direct networks which are initiated directly or triggered from other wireless technologies such as 4G/5G or Bluetooth using mobile devices. It is important to emphasize that the operational characteristics of Wi-Fi Direct network differ from infrastructure wireless LANs. Thus, an algorithm that suits one may not suit the other. In infrastructure wireless LANs, standalone and controlled access points can run computationally expensive algorithms since they are equipped with powerful internal circuits and have no power constraints and hence the algorithms proposed in [19–33] may work efficiently on them. On the other hand, Wi-Fi Direct is used in mobile handheld devices with modest internal circuits that have limited computational capabilities and power constraints. Therefore, we take into consideration, in designing the algorithm, not to make the mobile device perform intensive computations and run complex optimization algorithms that may overwhelm the device and affect its performance. Furthermore, the standalone access points are provided with channel scanning capabilities and the controlled access points which share the same controller are aware of all the activities of each other and therefore they can be equipped with optimization

and load balancing algorithms such as the ones proposed in [19–23, 25–27, 30, 33], while for Wi-Fi Direct networks, the surrounding environment is completely unknown and no possible information exchange can occur with other co-located networks. Therefore, in our design, the Wi-Fi Direct network makes decisions to enhance its performance based on its own experience and without getting feedbacks from any other co-located networks. Moreover, infrastructure wireless LANs usually last for long periods of time such as several days, weeks or months without even being reset. An algorithm which enhances the performance of these networks by adaptively interchanging channels such as the algorithm introduced in [33] can provide stability if the time gap between a channel change and the other is large. Unfortunately, this is not applicable for pop-up Wi-Fi Direct networks since they last only for few minutes or maximum few hours in some scenarios. Therefore, we consider in our design to enhance the performance of these networks within their short lifetimes. The stability is assured in our design since the network does not keep moving between channels in a heavily congested band. In case all channels in the frequency band are experienced and found congested, the network settles down in the least congested channel for the rest of the connection time. We follow a simple linear regression approach to measure the trend of congestion for each experienced channel. In spite of the fact that our algorithm works at the MAC level, no changes of the functionality of the CSMA/CA access mechanism are needed and no standardization efforts are required. Only some modifications shall be applied on the driver level of the wireless adapters. The proposed algorithm is completely independent of the specifications of the PHY layer and therefore it can be deployed on any IEEE802.11 physical. Moreover, it is completely interoperable with all versions of the IEEE802.11 family of standards. The functionality of the proposed algorithm does not affect the normal operation of any other wireless networks using IEEE802.11 standard, on the contrary, it enhances their performance implicitly by keeping pop-up Wi-Fi Direct networks away from the already congested channels, whenever possible. We perform practical prototyping of our algorithm by using the software driver of wireless adapters based on RTL8188CUS chipset in a Linux environment. We also perform simulations using NS-2 [40] to evaluate the performance of Wi-Fi networks in dense deployments.

The proposed algorithm considers channel assignments of pop-up Wi-Fi Direct networks in an environment where other networks belong to the same or different users or enterprises with open or private accesses. This is very important to consider since the majority of deployed wireless networks in all their modes and setups are actually private and protected by passwords especially the ones

implemented at homes, small/medium offices and commercial buildings. Therefore, in our algorithm, the users can only associate with their authorized networks. The algorithm does not consider fairness from the point of view of channel distribution between wireless networks since different infrastructure, ad-hoc and Wi-Fi Direct networks owned by same or different enterprises have different number of wireless stations, which is very typical in real-life wireless scenarios and thus the channels cannot be distributed equally among wireless stations to achieve fairness practically. Instead, in our algorithm, the channels are dynamically assigned to pop-up Wi-Fi Direct networks based on the real-time experience of each network alone to survive among the dense existence of other networks and therefore enhance their performance in a non-cooperative manner as mentioned before. The long term fairness in individually accessing the channel by each wireless station of the different networks is still maintained since it is a sole property of the CSMA/CA of the IEEE 802.11 distributed coordination function (DCF).

A thorough overview of Wi-Fi Direct's functionalities can be found in [41].

### 3 Proposed algorithm

#### 3.1 General description

The proposed algorithm abstractly works as follows. We choose network performance metrics for changing the operating channel and set thresholds accordingly. When one of the thresholds is reached, the pop-up Wi-Fi Direct network leaves its operating channel to another non-overlapping channel where it may find a better operating environment. The algorithm keeps running while the network is in operation. The decision of switching is taken by a wireless station involved in the communication process when it records a value of one of the chosen performance metrics which hits the chosen threshold of that metric. The station then processes a channel switch announcement to inform others with the new channel number. Stations switch their operating channel to the newly announced channel. If the whole band is heavily congested and the network has experienced all channels of the band, it settles down in the least congested channel for the rest of the connection time. The level of congestion for each channel is estimated using a linear regression approach.

Since our algorithm behaves in a non-cooperative manner, there is no feedback or coordination between the competing networks as mentioned before. The network has to decide by itself and based on its performance to exchange channels or to stay operating on the same channel. Therefore, the performance metrics shall be chosen to

have a direct reflection of channel congestion level from the network's point of view and the real need for leaving a channel to enhance performance.

#### 3.2 Performance metrics

The three performance metrics usually adopted to study the network performance from the MAC layer perspective are throughput, MAC delay and packet loss rate. Throughput is defined as the number of bits that is received successfully by the MAC layer of the receiver in one second. MAC delay is defined as the delay that a packet encounters from the time it enters the MAC queue until it reaches its destination and then being acknowledged successfully or dropped from the queue. Finally, packet loss rate is defined as the number of packets that have been lost from the overall transmitted packets. All of them have to be considered and investigated for possible selections as metrics for interchanging channels to enhance performance.

Realizing the average throughput as a metric is very practical since the number of packets that are being successfully received within a time period is directly affected by the number of competing stations operating on the same channel. As the number of wireless stations associated with the same network or other different networks increases, the possibility to find the channel busy by a source station is more probable and hence no transmission occurs unless the channel is sensed idle. This leads in turn to a reduction in the number of successful packets that is received at a time period by the receiver. It is worth mentioning that the obtained throughput by a wireless station may be affected by a factor other than channel congestion. This factor is the transmission rate. When stations emit their data on the channel at lower transmission rates rather than the nominal using different modulation and coding schemes (MCSs), its obtained throughput is reduced whether the channel is congested or not. Transmission rate factor can be avoided by normalizing the obtained throughput to the transmission rate of the source. Hereafter, the average normalized throughput is taken as one of the metrics for interchanging channels.

On the other hand, the average MAC delay to get the packet received successfully or dropped from the MAC queue is also directly affected by congestion. The station encounters delays when the channel is sensed busy. In spite of the fact that a MAC delay of a packet is also affected by the transmission rate of the station, this effect can be eliminated by calculating the MAC delay without taking the transmission time of a packet as part of the calculation. It can be achieved practically since the transmission time of a packet can be easily calculated by a station using both its current transmission rate and packet size. Thus, an



average customized MAC delay is chosen as another metric for interchanging channels.

Finally, the main sources of packet loss are the collisions that occur between packets sent by different wireless stations at the same time. With the increased number of competing stations which try to access the channel at the same time, the possibility of collisions increases. Therefore, collision rate is also a direct measure of congestion and consequently it is considered as a metric for interchanging channels.

All the three metrics are correlated. In other words, if one of them is affected by congestion in the surrounding environment, the others are also affected. Yet, we take in the proposed algorithm all the three metrics into consideration and not only one because usually the communication requirements of various types of applications and traffics are focused on one metric or more. The minimum acceptable value(s) of the metric(s) may be given or determined to serve the application efficiently. Therefore, our design is general for all types of applications using Wi-Fi Direct networks and the values of the thresholds can be customized according to the application requirements. For example, when using Voice over IP (VOIP), MAC delay as well as collision rate are important while throughput is not as much. Thus, the former thresholds can be set to meet the requirements of the used VOIP codec scheme, then the latter threshold is set accordingly. The thresholds can be considered by vendors and adopted with their correlations as adjusted variables in the settings page of the wireless drivers of the Wi-Fi Direct capable devices.

### 3.3 Detailed description and implementation

To implement our algorithm at the MAC level, we use counters and variables defined by the IEEE802.11 and explicitly or implicitly applied in the certified IEEE802.11 wireless devices of different vendors with different names and notations in order to estimate the normalized throughput, customized MAC delay and the collision rate. In describing the algorithm, we use the standard's naming and notations to generalize the use of the algorithm in any certified Wi-Fi Direct capable device and hence not to stick with specific vendor's implementation of the IEEE802.11 standard. The used counters and variables are discussed in the following.

Every wireless station maintains a station short retry count (*SSRC*) and a station long retry count (*SLRC*) with an initial value equals zero. The *SSRC* is incremented with every failed transmission attempt of a data packet whose length is less than or equal to *dot11RTSThreshold* and the *SLRC* is incremented with every failed transmission

attempt of a data packet whose length is greater than *dot11RTSThreshold*. Both *SLRC* and *SSRC* reset to zero in the following two cases. Either the number of retry limit of a packet reaches its maximum value; *dot11LongRetryLimit* for the former and *dot11ShortRetryLimit* for the latter or when the packet is successfully transmitted. Another counter defined by the standard that increments after each successfully transmitted packet is denoted by *dot11TransmittedFrameCount*. Moreover, a counter that increments after each packet is dropped from the queue, because either *dot11LongRetryLimit* or *dot11ShortRetryLimit* is hit, is denoted by *dot11FailedCount*. All of the aforementioned counters and variables run on the transmitting station. Thus, the transmitting station is fully aware of the conditions of all sent packets. Note that an error may occur to a transmitted packet either because of a collision or channel impairments. The counters *SLRC*, *SSRC* and *dot11FailedCount* do not differentiate between the sources of errors and hence they increment either way. Usually, the devices involved in a pop-up Wi-Fi Direct network are in a close proximity of each other and hence the effect of channel impairments is very small when compared with collisions and can be neglected as a source of error. Therefore, although the following equation represents the error rate, we intuitively denote it by collision rate only to magnify the fact that collisions are the dominant cause of errors in such networks. Hence, collision rate can be estimated by

$$\zeta = \frac{\sum_{i \neq j} SSRC_i + \sum_{i \neq j} SLRC_j}{\text{dot11TransmittedFrameCount} + \sum_{i \neq j} SSRC_i + \sum_{i \neq j} SLRC_j}, \quad (1)$$

where  $i, j$  are integers that represent packets' numbers.

To calculate the MAC delay of a packet, we tag the time at which the packet is either successfully received and acknowledged or dropped from the queue after several transmission attempts. Hence, we tag the times at which either *dot11TransmittedFrameCount* or *dot11FailedCount* is incremented. MAC delay of a packet is the difference between two successive time tags. The customized MAC delay is calculated by subtracting the transmission time from the obtained MAC delay.

Finally, the normalized throughput is calculated as follows. The sender adds the sizes of the packets which are only acknowledged by the receiver within a window of transmitted packets. This window shall represent an enough sample. We consider a window of 1000 packets in our experiments as discussed in Sect. 4. The sender tags the time at the beginning of the window and the time at the end and then subtracts them. The normalized throughput is obtained by dividing the calculated sum of packet sizes over the time difference multiplied by the data rate. The

window size is taken the same for calculating both, the collision rate and the normalized throughput.

Figure 1 shows the pseudocode of the algorithm. It is explained in the following. After a wireless adapter initiates a Wi-Fi Direct network, a counter  $i$  representing the number of transmitted packets is initiated to zero. Flag  $RF$  which represents the network experience in operating at all channels of the band more than once, is also initiated to zero. Moreover, the customized delays, average normalized throughputs and collision rates of all channels and their summations are initiated to zeros since no channels are experienced yet.

When stations get involved in a Wi-Fi Direct communication, the wireless station saves the current time in  $T_{old\_DL}$ ,  $T_{start\_TH}$  which are used in calculating the delay and throughput respectively and initializes the unsuccessful transmissions counter  $US$ . Then, it starts monitoring the transmitted packets. The station tags the time at which either  $dot11TransmittedFrameCount$  or  $dot11FailedCount$  are incremented and save it in  $T_{new\_DL}$ . A flag  $TF$  is incremented as well. On the other hand, if neither of the two counters is incremented, the same packet is retransmitted according to the standard and a counter  $RT$  is incremented instead to count the number of retransmission attempts. In the latter case, the value of the  $SLRC$  or  $SSRC$  is stored in  $FRC$ .  $US$  is incremented by the value of  $FRC$  to calculate the collision rate at a later stage. If  $dot11TransmittedFrameCount$  is incremented, the size of the packet is added to the bit accumulator of the successfully transmitted packets  $PSA$  to calculate the throughput at a later stage as well. The MAC delay  $PEED$  is calculated for each packet without considering the transmission time of all attempts. The customized average MAC delay  $AEED$  of all sent packets is calculated according to the recorded  $T_{old\_DL}$  and  $T_{new\_DL}$ .

When either, counter  $i$  hits  $CP$  or no more pending packets are in the MAC queue, the station calculates the normalized average throughput and stores it in  $NT$  after tagging the current time  $T_{end\_TH}$ . It also calculates the collision rate and stores it in  $CR$ . The station then compares the three metrics with their preset thresholds  $EEDT$ ,  $CRT$  and  $THT$ . The station remains performing the previous operations as long as the average customized MAC delay and the collision rate are less than their thresholds and the normalized average throughput is greater than its threshold. When one or more of the three thresholds are hit and the number of transmitted packets is  $CP$  or more, the station starts the procedure of changing its operating channel  $RC$ . During this procedure and right before any switches occur, the station keeps a record of the values of the three metrics that is experienced at each channel. It stores these values in their associated arrays,  $AEED\_LIST(RC)$ ,  $CR\_LIST(RC)$  and  $NT\_LIST(RC)$ . The elements of the array of each

metric for all experienced channels is accumulated and stored in  $AEED\_ALL$ ,  $CR\_ALL$  and  $NT\_ALL$  for later use.

Channel switching may occur in two stages. The first stage is when the network has not experienced all channels of the frequency band. The second stage only happens when all channels of the frequency band have been experienced at least once because of a substantially congested band. In the first stage, the station randomly selects a non-overlapping channel in the operating frequency band as long as it has not experienced that channel before. Therefore, after each channel switching, the old channel number is removed from the list of the channel numbers  $AC$ . Note that the station switches channels after it operates at each channel and one or more of the three metrics hit their preset thresholds. The station resets  $i$ ,  $AEED$ ,  $NT$  and  $CR$  after each channel switch. If all channels are experienced,  $AC$  gets empty and hence  $RF$  flag is set to one. The station performs the second stage of channel switching when  $RF$  is one and one or more of the three thresholds are hit after experiencing all channels in the frequency band and no more channels are left to experience. Note that the station may have left the experienced channels according to different metrics. For example, it may have left channel one because the throughput went below its preset threshold and it may have left the second channel because the collision rate went above its preset threshold and so on. The three metrics are considered by taking the following steps. First, weighing the value of every metric of each channel according to the accumulated values of that metric in all channels. Thus, the normalized throughput, average customized MAC delay, and collision rate of each experienced channel are divided by the summation of the normalized throughputs, average customized MAC delay, and collision rate, respectively, of all channels. This makes the results get represented as ratios rather than values with different units. Thus, the ratios can be added together for each channel and then stored in an array which is denoted by  $ALL$ . The index of this array is the channel number. Note that the average customized MAC delay and collision rate should be less than their preset thresholds while average normalized throughput should be above its preset threshold to maintain good performance. Hence, the calculated ratio of the throughput is subtracted from the calculated ratios of the other two metrics. The best channel to choose for the rest of the network lifetime, supposedly, is the one which points to the minimum value of the summed ratios in the array  $ALL$ . However, some values of the array may be close to each other because the values of thresholds are the same when the network operates in any channel which makes the station leave some channels with almost same or close averages. Hence, the minimum average, when near values occur around the minimum, may not give a clear indication about the right channel to use to achieve best performance.

```

Declare variable  $i$  // number of packets
Declare variable  $AC$  // list of channel numbers of the frequency band
Declare variable  $NC$  // highest channel number of the frequency band
Declare variable  $CCN$  // current channel number
Declare variable  $RC$  //new channel number
Declare variable  $TF$  //successful or dropped packet flag
Declare variable  $RF$  // round 1 Flag
Declare variable  $RT$  // retransmission attempts of a packet
Declare variable  $CP$  // enough sample of packets
Declare variable  $ALL$  // array of weighted metrics of all channels
Declare variable  $PSA$  // accumulated sizes of successfully sent packets
Declare variable  $NT$  // average normalized throughput
Declare variable  $NT\_LIST$  //array of normalized throughput of all channels
Declare variable  $NT\_ALL$  // accumulated throughput of all used channels
Declare variable  $THT$  // throughput threshold
Declare variable  $T_{start\_TH}$  // time tag for throughpt calculation purposes
Declare variable  $T_{end\_TH}$  // time tag for throughput calculation purposes
Declare variable  $US$  //unsuccessful transmissions counter
Declare variable  $CR$  // collision rate
Declare variable  $FRC$  // store the value of the SLRC or SSRC
Declare variable  $CRT$  // collision rate threshold
Declare variable  $CR\_LIST$  // array of collision rate for all channels
Declare variable  $CR\_ALL$  // accumulated collision rate of all used channels
Declare variable  $PEED$  // packet MAC delay
Declare variable  $AEED$  // average MAC delay
Declare variable  $AEED\_LIST$  // array of MAC delays for all channels
Declare variable  $AEED\_ALL$  // accumulated delay of all used channels
Declare variable  $EEDT$  // MAC delay threshold
Declare variable  $T_T$  // transmission time
Declare variable  $T_{old\_DL}$  // time tag for delay calculations
Declare variable  $T_{new\_DL}$  // time tag for delay calculations
Declare variable  $T_{new\_TS}$  // time tag for the calculations of trendline's slope
Declare variable  $T_{old\_TS}$  // time tag for the calculations of trendline's slope
Declare variable  $TS$  // array of trendlines' slopes of all experieced channels
Declare variable  $T_{int\_TS}$  // preset time slot for trendline's slope calculations
Declare variable  $C_{TS}$  // integer increments of  $T_{int\_TS}$ 
Declare variable  $EDTS$  // array of MAC delay at  $C_{TS}$  increments

while wireless adapter is connected and on
   $ST \leftarrow \text{dot11TransmittedFrameCount}$ 
   $FT \leftarrow \text{dot11FailedCount}$ 
   $RF \leftarrow 0$ 
   $i \leftarrow 0$ 
   $AEED\_ALL \leftarrow 0$ 
   $NT\_ALL \leftarrow 0$ 
   $CR\_ALL \leftarrow 0$ 
  for  $j = 1$  to  $NC$  do:
     $AEED\_LIST(j) \leftarrow 0$ 
     $CR\_LIST(j) \leftarrow 0$ 
     $NT\_LIST(j) \leftarrow 0$ 
  endfor
  while station belongs to Wi-Fi Direct Network
     $T_{old\_DL} \leftarrow \text{get current time}$ 
     $T_{start\_TH} \leftarrow \text{get current time}$ 
     $T_{old\_TS} \leftarrow \text{get current time}$ 
     $C_{TS} \leftarrow 0$ 
     $US \leftarrow 0$ 
    while transmission queue has pending packets
       $i \leftarrow i + 1$ 
       $TF \leftarrow 0$ 
       $RT \leftarrow 0$ 
      while  $TF=0$ 
        transmit packet
        if  $ST$  is incremented
           $PSA = PSA + \text{PacketSize}$ 
        endif
        if  $ST$  is incremented OR  $FT$  is incremented
           $T_{new\_DL} \leftarrow \text{get current time}$ 
           $TF \leftarrow 1$ 
        else
           $RT \leftarrow RT + 1$ 
          if  $\text{PacketSize} > \text{dot11RTSThreshold}$ 
             $FRC = \text{SLRC}$ 
          else
             $FRC = \text{SSRC}$ 
          endif
        endif
      endwhile
    endif
     $US = US + FRC$ 
     $T_T \leftarrow \text{PacketSize}/\text{DataRate}$ 
     $PEED \leftarrow (T_{new\_DL} - T_{old\_DL} - RT * T_T)$ 
     $AEED \leftarrow (AEED*(i-1) + PEED)/i$ 
     $T_{old\_DL} \leftarrow T_{new\_DL}$ 
     $T_{c\_new} \leftarrow \text{get current time}$ 
    if  $T_{new\_TS} \geq T_{old\_TS} + T_{int\_TS}$ 
       $C_{TS} = C_{TS} + 1$ 
       $EDTS(C_{TS}) \leftarrow PEED$ 
       $T_{old\_TS} \leftarrow T_{new\_TS}$ 
    endif
    if ( $i >= CP$ )
      exit
    endif
  endwhile
   $CR \leftarrow US / (ST + US)$ 
   $T_{end\_TH} \leftarrow \text{get current time}$ 
   $NT = (PSA / (T_{end\_TH} - T_{start\_TH}))/\text{DataRate}$ 
  if ( $i \geq CP$ ) AND ( $AEED \geq EEDT$ ) OR ( $CR >= CRT$ )
    OR ( $NT \leq THT$ )
     $CR\_LIST(RC) \leftarrow CR$ 
     $CR\_ALL = CR\_ALL + CR$ 
     $NT\_LIST(RC) \leftarrow NT$ 
     $NT\_ALL = NT\_ALL + NT$ 
     $AEED\_LIST(RC) \leftarrow AEED$ 
     $AEED\_ALL = AEED\_ALL + AEED$ 
     $CCN \leftarrow \text{get current channel number}$ 
    for  $z=1$  to  $C_{TS}$ 
       $TS(CCN) \leftarrow \text{Calculate trendline slope of EDTS}(z)$ 
    endfor
    if  $RF = 0$  AND  $AC \neq \Phi$ 
       $RC \leftarrow \text{randomly select channel from AC}$ 
       $AC \leftarrow AC - RC$ 
      if  $AC = \Phi$ 
         $RF \leftarrow 1$ 
      endif
      process channel switch announcement
      switch to channel  $RC$ 
    else
      for  $j = 1$  to  $NC$  do:
         $ALL(j) = [(AEED\_LIST(j)/AEED\_ALL) + (CR\_LIST(j)/CR\_ALL) - (NT\_LIST(j)/NT\_ALL)]$ 
        for  $k = j+1$  to  $NC$  do:
           $ALL(k) = [(AEED\_LIST(k)/AEED\_ALL) + (CR\_LIST(k)/CR\_ALL) - (NT\_LIST(k)/NT\_ALL)]$ 
          if  $ALL(k) < ALL(j)$ 
             $RC \leftarrow k$ 
          endif
          if ( $ALL(k) / ALL(j)$  OR  $ALL(j) / ALL(k) < 10\%$ )
            AND ( $TS(j) < TS(k)$ )
             $RC \leftarrow j$ 
          endif
        endfor
      endfor
      if  $RC = CCN$ 
        stay in channel  $CCN$ 
      else
        process channel switch announcement
        switch to channel  $RC$ 
      endif
    endif
     $i \leftarrow 0$ 
     $AEED \leftarrow 0$ 
     $NT \leftarrow 0$ 
     $CR \leftarrow 0$ 
  endif
endwhile

```

Fig. 1 Algorithm pseudocode



Therefore, when the low values of the array *ALL* are within certain percentages of each other, a simple approach to consider is to check the trend of congestion in each channel derived from the obtained results of the station when it experienced that channel. A simple linear regression approach is taken as follows. The slope of a linear trendline of the results of a preselected metric of any of the three metrics is calculated for each channel when the station is about to perform a switching operation and right before it leave its operating channel in the first switching stage. In the second switching stage and while determining the minimum value of the array *ALL*, only the channels which have near values around the minimum are further investigated by considering their calculated trendline slopes of the selected metric. From these channels, the channel with the highest corresponding slope is chosen as the next and last operating channel if the used metric is the throughput or else the channel with the lowest corresponding slope is chosen if one of the other two metrics are used instead. It does not matter which metric is chosen since the slopes of the three metrics are correlated and will lead to the same result. We consider the data stored in the array *ALL* close if their values are 90 % correlated. In Fig. 1, we calculate the slopes of the trendlines of the customized MAC delay which are stored in the array *EDTS* for each experienced channel and put them in an array *TS*. To reduce the computational cost of the slopes as well as the amount of stored data in the array *EDTS*, a preset time interval  $T_{int\_TS}$  is defined and used in the process of calculating the slopes. Hence, only partial results of the obtained delay values are considered in the calculations which are enough to give an indication on the trend of congestion.

As mentioned before, if the whole band is congested and the station finally chooses the least congested channel, it does not keep switching channels during a network session. Again, we emphasize the fact that unlike infrastructure IEEE802.11 networks, a pop-up Wi-Fi Direct network is a temporary network which is mostly being established between devices for short periods of time. Therefore, it is realistic to make a network experience each channel once, only if needed and according to the practiced performance. Then, if and only if all channels are experienced, the network settles down and stays operating at the least congested channel. This assures the stability operation of the network.

The values of the preset thresholds of the average normalized throughput, average customized MAC delay and collision rate have to be carefully chosen to enhance the network performance, by switching channels, and at the same time stabilize the operation of the network. Therefore, it is important to study the characteristics of these metrics carefully to provide a reference for choosing their preset thresholds.

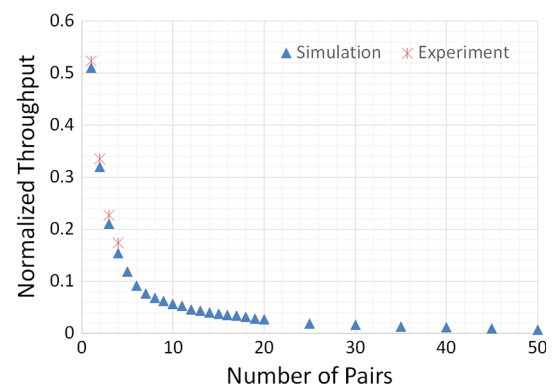
## 4 Performance evaluation

### 4.1 Simulation results

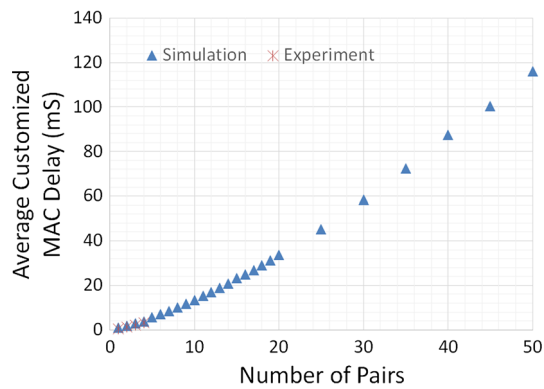
We first perform simulations using NS-2 to evaluate the network performance in dense deployment. We also perform experiments with few number of devices to validate the simulation results. We then use the obtained results of the simulations in choosing proper values of the preset thresholds of the algorithm.

In the simulations, the stations are grouped in pairs and set to compete on the same channel. Each communicating pair consists of one transmitter and one receiver and emulates one pop-up Wi-Fi Direct network. In the experiments, eight wireless USB adapters that support Wi-Fi Direct are used. Four pop-up Wi-Fi Direct networks are established. The default values of all the parameters of the DCF access mechanism are adopted. Some of them are mentioned in the following. The minimum contention window is 15, the maximum contention window is 1023, the short retry limit is 7, the long retry limit is 4, the *RTSThreshold* is 3000, the *FragmentationThreshold* is 2346, the *MaxReceiveLifetime* is 512 and the *MaxTransmitMsduLifetime* is 512. Greedy traffics are generated in both the simulations and experiments to illustrate the worst case scenario in dense deployments. Each measurement is attained after a test period of 280 s.

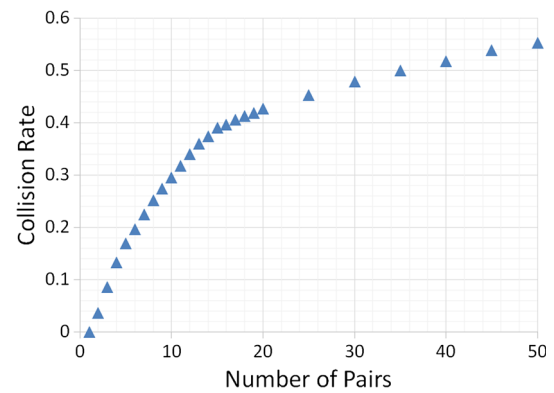
We first make one pair communicate on the channel and record the obtained average throughput. Then, we increase gradually the number of pairs. Figures 2 and 3 show the average normalized throughput and the average customized MAC delay respectively with the increasing number of pairs as found by both the simulations and experiments. As shown, the simulation results almost resemble the results obtained by the experiments. We also use simulation to find the average normalized throughput and the average customized MAC delay for scalable number of pairs. From Fig. 2, the average normalized throughput decreases with



**Fig. 2** Average normalized throughput as a function of the number of communicating pairs



**Fig. 3** Average customized MAC delay as a function of the number of communicating pairs



**Fig. 4** Collision rate as a function of the number of communicating pairs

the increasing number of pairs. The throughput exponentially decays since wireless stations perform the CSMA/CA with the exponential backoff algorithm which provides fairness between competing pairs. The average customized MAC delay, as mentioned earlier, is the average MAC delay excluding the transmission time. As shown in Fig. 3, it increases with the increasing number of competing pairs. The delay ranges from few milliseconds for a small number of pairs to the order of tens of milliseconds for a large number of pairs. The delay is large for a large number of pairs mainly because of the long waiting times the stations spend in the backoff stages waiting for an idle channel to emit their data. Figure 4 shows the collision rate with the increasing number of pairs. The collision rate increases with the increasing number of competing pairs. The amount of increase gets slighter when the number of pairs is large since the probability that a station senses the channel busy increases. From the figure, almost half of the transmission attempts fail due to collisions in a dense environment.

As illustrated in Figs. 2, 3 and 4, congestion due to many competing stations operating on the same channel severely affects the performance of the network which is directly reflected on the throughput, delay and collision rate.

## 4.2 Experimental measurement

We implement our algorithm in a Linux environment by utilizing the driver of the RTL8188CUS chipset. We conduct four experiments to evaluate our algorithm in different scenarios. For illustration purposes only, we consider in the experiments a pop-up Wi-Fi Direct network which is initiated to transfer files between devices. Since throughput is the key metric for this kind of transfer and to make a Wi-Fi Direct network not to obtain average throughput less than

10 % of its nominal and also not to make unnecessary transitions, the threshold is set to 0.051 which is obtained from Fig. 2. This corresponds to setting the threshold values of the average customized MAC delay and collision rate to around 15 ms and 31 % respectively. The enough sample of packets after which the station starts checking on the values is 1000. The slopes of the trendlines are calculated from a data set stored every 1 s.

To demonstrate the benefits of the proposed algorithm and to emphasize the fact that it can run on and improve the performance of any pop-up Wi-Fi Direct network operating in any environment, we run our algorithm in uncontrolled environments in the first two experiments. In these environments, many of users are accessing different types of IEEE802.11 networks and setups and they are owned by different people or different enterprises which we do not have access on. In these experiments, we evaluate the proposed algorithm in both a commercial and a residential buildings and provide detailed discussions of the obtained results. We also conduct the third and the fourth experiments to compare the proposed algorithm with other algorithms previously proposed in the literature [32, 33]. We concentrate in the last two experiments on comparing the algorithms in terms of performance and stability. In the fourth experiment, we compare our algorithm with the one proposed in [32] in an uncontrolled environment while we conduct the third experiment in a controlled environment that we setup to compare the algorithm with the one adopted by Aruba [33]. The choice of a controlled environment in the latter is only because we use an access point connected to a controller which both exist in the Communication Networks Laboratory of Princess Sumaya University for Technology.

In the first experiment, we run our algorithm in an uncontrolled dense environment in a small commercial building. Let us first consider the distribution of the Wi-Fi networks among the channels of both the 2.4 GHz and

5 GHz frequency bands in this building. Figures 5 and 6 show snapshots of the distributions of the Wi-Fi networks in the chosen environment in the 2.4 and 5 GHz bands respectively. The snapshots are taken using Wifi Analyzer [42] running on an Android-based smartphone. Figure 5 illustrates the heavy usage of Wi-Fi channels in the 2.4 GHz band while Fig. 6 shows that the usage of the channels of the 5 GHz band is not as much. The latter is less, since the coverage range in the 2.4 GHz band is much better than the coverage range in the 5 GHz band when using IEEE802.11b,g,n devices. IEEE802.11ac devices which operate only in the 5 GHz band are not spread yet. Most of the currently used Wi-Fi devices run the legacy IEEE802.11n. This is very typical in many residential and commercial buildings everywhere around the world. The use of IEEE802.11ac devices is expected to increase fast in the near future with their continuously decreasing prices. Without losing the generality, we only run the algorithm in the 2.4 GHz band since the used wireless adapters only operate in this band. Since the algorithm is implemented at the MAC level, we still emphasize the fact that the algorithm works with any PHY and in any band that Wi-Fi Direct may use.

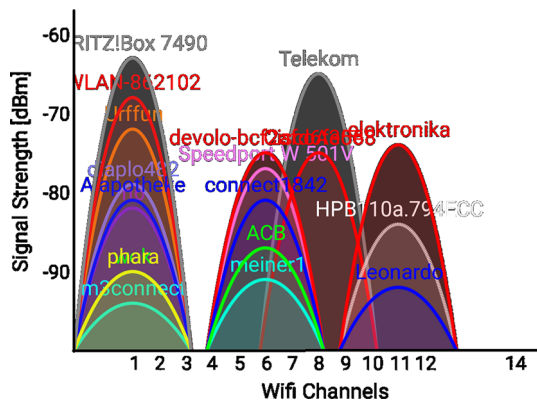


Fig. 5 The distribution of Wi-Fi networks operating in the 2.4 GHz frequency band in a commercial building

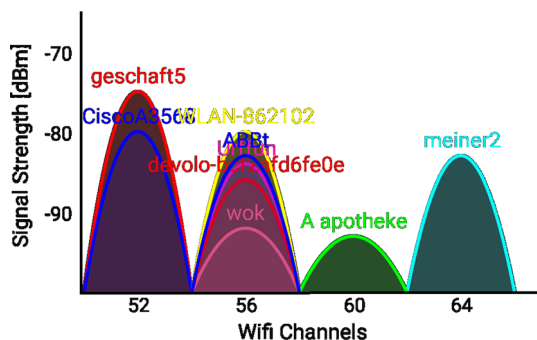


Fig. 6 The distribution of Wi-Fi networks operating in the 5 GHz frequency band in a commercial building

The modified network initially operates at channel one. The wireless adapter chooses this channel at the network establishment time according to the algorithms already implemented in the driver of the wireless adapter. Greedy traffic is generated for 280 s. Average normalized throughput, average customized MAC delay and collision rate are recorded in one second intervals. The obtained results of the average normalized throughput, the average customized MAC delay and the collision rate are shown in Figs. 7, 8 and 9 respectively. From the figures, the station operates in channel one for only a short period of time since it encounters throughput below its preset threshold and delay and collisions above their preset thresholds. Therefore, as soon as the number of packets reaches 1000, at around 20 s, the station leaves channel one to channel eleven. The average values of the normalized throughput, delay and collision rate at the time the station leaves channel one are 0.04132, 18.65 ms, 33.62 %, respectively. Note that all of the metrics exceeded their preset thresholds. The used wireless adapters take around 2 s to move the network to a new operating channel. The network then operates at channel eleven where the network gets better

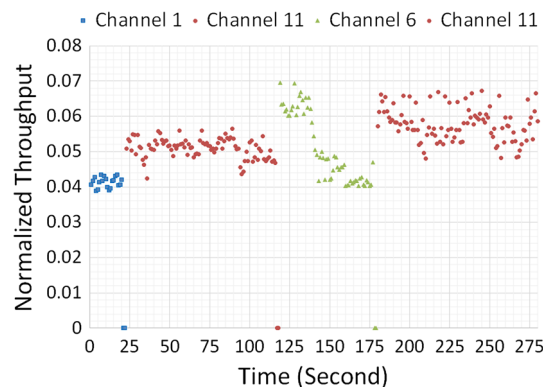


Fig. 7 Normalized throughput attained by the modified Wi-Fi Direct network running the algorithm in a commercial building

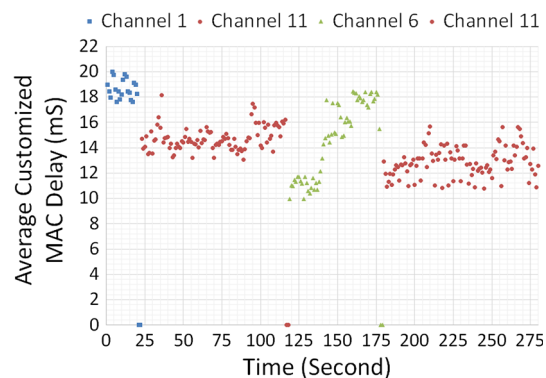
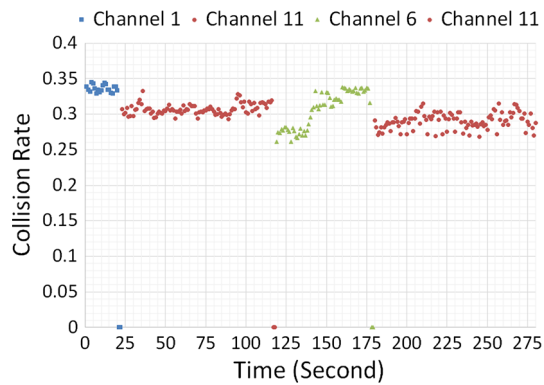


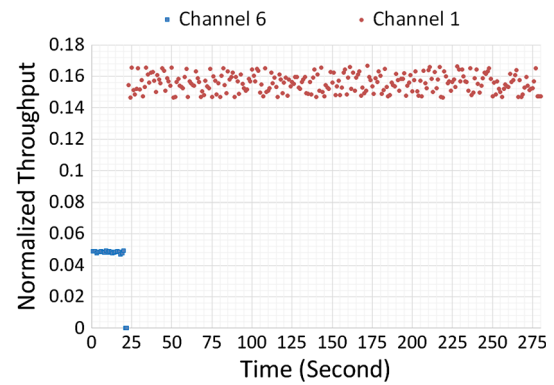
Fig. 8 Average customized MAC delay encountered by the modified Wi-Fi Direct network running the algorithm in a commercial building



**Fig. 9** Collision rate of the modified Wi-Fi Direct network running the algorithm in a commercial building

performance than channel one. Yet, the preset threshold is also met after operating in that channel for around 94 s. The average values of the normalized throughput, delay and collision rate at the time the station leaves channel eleven are 0.050999, 14.74 ms and 30.71 % respectively. Hence, the station leaves channel eleven and then operates at the remaining unexperienced non-overlapping channel, six. In the first few seconds, the station encounters good performance but after that the performance dramatically decreases, which makes the station leave the channel after 59 s. The average values of the normalized throughput, delay and collision rate at the time the station leaves channel six are 0.050993, 14.71 ms and 30.69 %, respectively. Hence, the station leaves channel six. By operating at channel six, the station experienced the three non-overlapping channels of the ISM band and hence the first switching stage is over. Henceforward, the station has to decide the best channel of the three to use in the remaining time of the connection. Note that the values of the three metrics when the station operates at channels six and eleven are very close. The difference in the summed weighted averages of the three metrics for these two channels are less than 10 %. Therefore, the station compares the slopes of the trendlines of the obtained results of the delay for these two channels. The slope of the trendline is 0.0093 for channel eleven while it is 0.1558 for channel six. Hence, the network chooses channel eleven in the second switching stage as the last operating channel.

The normalized throughput increased by 39.97 % while the average customized MAC delay and the collision rate decreased by around 31.11 and 13.74 %, respectively when the station finally operates at channel eleven rather than its few seconds operation at channel one. On the other hand, the normalized throughput increased by 13.42 % while the delay and the collision rate decreased by around 12.7 and 5.52 %, respectively, when the station finally operates at channel eleven rather than its last few seconds operation at channel six.

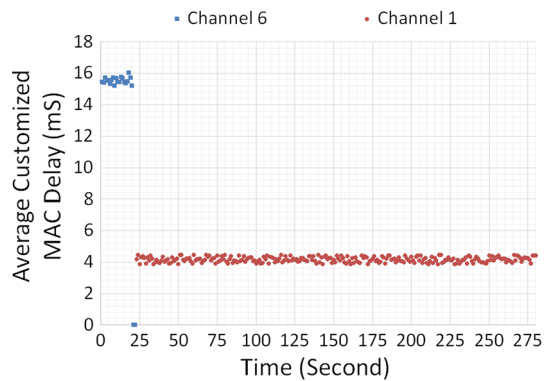


**Fig. 10** Normalized throughput attained by the modified Wi-Fi Direct network running the algorithm in a residential building

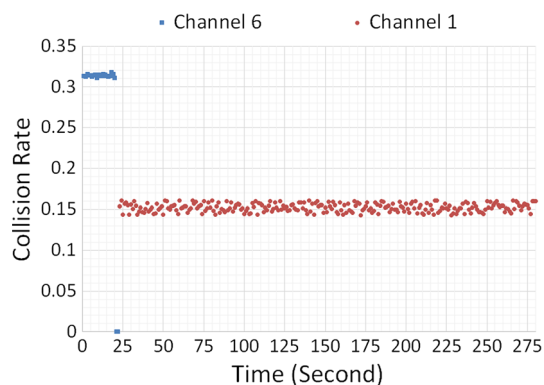
Note that in commercial buildings, every network administrator of an office or a company monitors the performance of their private Wi-Fi networks and tries to avoid the channel overlapping with other networks as much as possible. In spite of this fact, the algorithm enhanced the performance of the modified pop-up Wi-Fi Direct network in such environment. We performed the experiments in such environment to ensure the effectiveness of using the proposed algorithm even in an environment with wise management. In residential buildings, regular people usually do not keep an eye on their networks especially with their limited knowledge of network handling and management. In residential buildings, practically, some channels are occupied by many overlapped networks owned by different residents while other channels are occupied by few networks or they are even completely empty. We consider this kind of environment by performing the second experiment in a residential building. Same procedure as in the previous experiment is followed. The modified network initially selects channel six as the operating frequency. The values of the three metrics are recorded. They are shown in Figs. 10, 11 and 12. From the figures, the station operates at channel six for only a short period of time. It randomly selects channel one as soon as the number of packets reaches 1000. The network stays operating at this channel for the rest of the connection time, where it encounters a huge improvement in its performance. The normalized throughput increased by 221 % while the delay and the collision rate decreased by around 73.16 and 51.47 %, respectively.

We now compare our algorithm by the one adopted by Aruba. The latter is a part of the adaptive radio management (ARM) technique implemented in Aruba's access points which are managed by a controller [33]. One of ARM's features is the ability of a network to move between channels when errors occur in the transmitted packets. This feature, in principle, is analogous to our proposed algorithm with only one metric is adopted instead of three as in ours at the MAC level. It is important to





**Fig. 11** Average customized MAC delay encountered by the modified Wi-Fi Direct network running the algorithm in a residential building



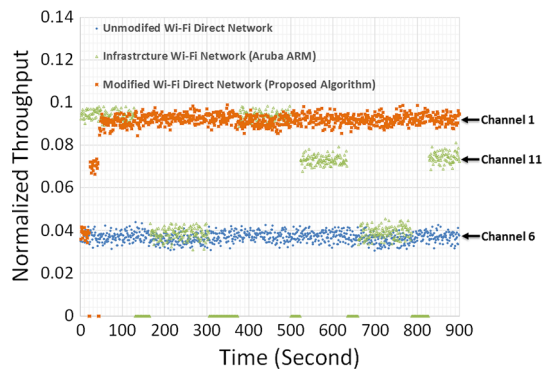
**Fig. 12** Collision rate of the modified Wi-Fi Direct network running the algorithm in a residential building

emphasize that ARM is specifically designed for IEEE802.11 infrastructure mode and therefore it provides many enhanced functionalities and features that suit these infrastructure managed networks while our proposed algorithm is specifically designed for Wi-Fi Direct which has different operational characteristics as explained earlier. Nevertheless, we perform the experiments in an operational environment that suits Wi-Fi Direct to show the advantage of using the proposed algorithm over the one adopted by ARM especially in terms of stability. It is worth mentioning that Aruba ensures the stability of the operation of its infrastructure managed networks in not to keep moving between channels so often by setting a high default value for the error rate threshold which equals 50 %. It also sets a timer, denoted by error rate wait time, which counts the minimum time in seconds the error rate has to exceed the threshold before it triggers a channel change. The default value of this timer is 30 s. Moreover, after an access point changes a channel, it has to wait a time interval, denoted by backoff time, before it asks the controller for a new channel. The range of possible values of the latter is 120–3600 s with a default value equals 240 s.

The conducted third experiment is as follows. We create twenty-six Wi-Fi Direct networks and one infrastructure Wi-Fi network in the Communication Networks Laboratory of Princess Sumaya University for Technology with the help of the communications and electronics engineering students. The mostly used devices in the experiment are smartphones while the rest are PCs and laptops. Each Wi-Fi Direct network consists of a transmitter and a receiver. Six devices are connected to the access point of the infrastructure network. We also create one Wi-Fi Direct Network that runs the proposed algorithm which also consists of a transmitter and a receiver. Moreover, we create an infrastructure Wi-Fi network using an Aruba controller with model number 3200XM and an access point with model number AP125. We connect only one wireless device to the access point of the latter network and put them in a close proximity of each other to dilute the effect of channel impairments as a source of packet loss. Hence, collisions are the main source of errors in the transmitted packets. To provide a dense environment with many active users, greedy traffic is generated in all the involved networks during the whole experiment period. We run the experiment for 900 s. The thresholds of the normalized throughput, average customized MAC delay and collision rate are set in the modified network running the proposed algorithm to 0.1, 6.3 and 18 %, respectively. Using the Aruba controller setting page, we set the error rate threshold, error rate wait time and backoff time to 18 %, 1 and 120, respectively. The chosen settings suit the operational characteristics of a pop-up Wi-Fi Direct network with a short lifetime where decisions have to be made fast to enhance its performance in a dense deployment while in the same time keeping it stable.

Figure 13 shows the attained throughputs of (a) the modified Wi-Fi Direct network running the proposed algorithm, (b) an unmodified Wi-Fi Direct network and (c) the Aruba network running ARM. The figure shows that the modified network initially operates at channel six where its preset thresholds are hit as soon as the number of transmitted packets reaches 1000. It randomly selects channel eleven where its preset thresholds are also hit. It then moves to operate at channel one. In spite of the fact that the preset thresholds are also hit at channel one, it keeps operating at this channel for the rest of its lifetime since it is the least congested channel. The figure also shows that Aruba network initially operates at channel one then it keeps moving between channels during the whole lifetime of the connection since the threshold is always exceeded in all channels. It stays operating at each channel around 2–3 min. The time it takes Aruba network to leave a channel and move to another one ranges from 23 to 70 s. Figure 13 also shows that the unmodified Wi-Fi Direct network with the fixed channel allocation strategy of the





**Fig. 13** Normalized throughputs attained by the modified Wi-Fi Direct network running the proposed algorithm, an unmodified Wi-Fi Direct network and Aruba infrastructure network running ARM

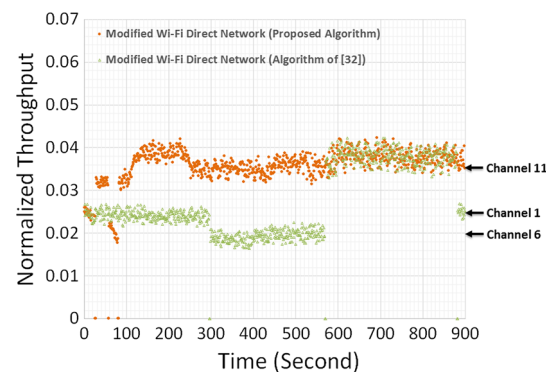
standard operating at channel six achieves the lowest throughput since this channel is the mostly congested channel in the band. Hence, it can be deduced from the obtained results that the proposed algorithm outperforms Aruba's algorithm in both performance and stability for networks with short lifetimes such as pop-up Wi-Fi Direct networks.

Finally, we implement the algorithm of [32] in a Linux environment by also utilizing the driver of the RTL8188CUS chipset. Using this implementation, a fair comparison can be provided between the two algorithms since two Wi-Fi Direct networks are now established, one which runs the proposed algorithm and another one which runs the algorithm of [32]. An abridged description of the algorithm of [32] is as follows. A network starts operating using a channel in the 2.4 GHz frequency band, when it encounters a number of lost packets beyond a preset threshold value, it leaves that channel and moves to the next channel in the band, in order. It keeps performing the algorithm continuously and switches channels without stopping as long as the threshold of the performance metric is hit. The adopted performance metric is *dot11FailedCount* which was described earlier in Sect. 3.3.

In this experiment, we set the threshold value of *dot11FailedCount* to 100, as recommended in [32], for the Wi-Fi Direct network running the algorithm of [32] and adopt the threshold value which we used in the first two experiments for the modified Wi-Fi Direct network running our proposed algorithm. Since, we concentrate on comparing the two algorithms in terms of performance and stability, we perform the experiment in a dense deployment where the threshold values are always exceeded. Figure 14 shows the attained throughputs by the two networks. The figure shows that the modified network running the proposed algorithm initially operates at channel one. It then leaves channel one to channel eleven, then to channel six and finally back to channel eleven, for the rest of the

connection time, wherein it experiences the best performance. The figure also shows that the modified network running the algorithm of [32] initially operates at channel one. It then moves between the three non-overlapping channels of the band, in order, during the connection time. Although channel eleven is less congested than channel one, the network leaves channel eleven back to channel one, which was previously experienced, after 881 s from the beginning of the connection time. This is because the algorithm of [32] does not include a mechanism that selects the least congested channel after experiencing them all and it keeps moving between channels, in order, as soon as the threshold value is hit during the connection time.

It is important to clarify that the algorithm of [32] considers *dot11FailedCount* without considering the number of all transmitted packets. In other words, it considers an absolute number of the packet loss and not the rate of packet loss. So even if the number of successfully transmitted frames is large and way exceeds the number of failed frames, the algorithm makes a network leave a channel as soon as the threshold of *dot11FailedCount* is hit. Since eventually by time, *dot11FailedCount* reaches the preset threshold value, even with a slightly used channel, the network keeps switching channels, without stopping, which affects the stability of the network. The only way to reduce the number of switches during a connection time and prolong the operation of the network in each channel is by choosing a high threshold value such as 100, which is recommended in [32] as we mentioned before. This explains why in the performed experiment, the modified network running the algorithm of [32] only moved three times between the channels during the connection time. On the other hand, setting a high threshold value affects the performance of the network since it may stay operating in a congested channel for a considerable amount of time before the threshold value is reached. This explains the bad performance experienced by the modified



**Fig. 14** Normalized throughputs attained by the modified Wi-Fi Direct network running the proposed algorithm and the modified Wi-Fi Direct network running the algorithm of [32]

network running the algorithm of [32] in the conducted experiment as illustrated in Fig. 14. As for the algorithm of [32], it is a matter of trading off between stability and performance. Hence, it can be deduced from the conducted experiment that the proposed algorithm outperforms the algorithm of [32] since it makes a balance between stability and performance as well as it has a mechanism to select the least congested channel when all channels are experienced.

## 5 Conclusion

In this paper, we proposed an algorithm that enhances the performance of the very promising technology, Wi-Fi Direct in dense environment. The algorithm is especially designed for pop-up Wi-Fi Direct and suits the behavior of such networks. We also discussed the importance of Wi-Fi Direct as being the possible tentative for high speed device-to-device communication in smart cities. The proposed algorithm only involves simple, yet effective, modifications in the software drivers of wireless adapters operating using Wi-Fi Direct. Therefore, no need for any MAC modifications within the standard and hence no standardization efforts are required. Moreover, our algorithm does not involve computationally expensive optimization techniques or algorithms which may overwhelm portable handheld devices. We targeted in our work to provide a practical algorithmic design with actual prototyping and implementations rather than just providing a pure theoretical optimized approach loaded with many assumptions which may throttle the overall performance when practically implemented. We actually utilized some counters of the IEEE802.11 standard in designing and implementing the algorithm. Using these counters, we estimate the throughput, delay and collision rate which are the most important MAC performance metrics. We adopted these metrics as the key metrics for interchanging channels. We also utilized the Channel Switch Announcement defined in the standard to make the sender inform the receiver to switch the operating channel when the preset threshold of any metric is hit. The thresholds can be adjusted by network administrators based on their desired acceptable performance. The stability of the network is assured since the network changes its operating channel only when it faces bad performance according to the preset thresholds. Furthermore, in some scenarios where the whole band is heavily used, the network settles down in the least congested channel. The trend of congestion of each channel is estimated by using a simple linear regression approach. We prototyped our work by modifying the source code of the software driver of wireless adapters based on RTL8188CUS chipset in a Linux environment. We used this open-source driver since it is provided under GNU General Public License ver.2 [43] and it strongly supports Wi-Fi Direct. Moreover, it is not part

of the Linux already compiled wireless drivers and hence its source code can be modified, compiled and then uploaded as a module to the kernel using modprobe command. It also has private command line instructions to modify the settings using iwpriv command.

The obtained results of the conducted experiments illustrate the increase in performance when using our adaptive algorithm in a dense environment over the fixed channel allocation strategy which is currently used by Wi-Fi Direct networks. The proposed algorithm can increase the performance of Wi-Fi Direct networks operating in some environments such as residential ones in the order of multiples. As was shown in the third experiment, when the error rate threshold was set to 18 % and heavy traffic was conducted for 15 min which is a typical lifetime for many Wi-Fi Direct applications but not for infrastructure managed networks, Aruba network kept moving between channels during the whole connection time which made the network unstable. Therefore, it can be deduced that Aruba's algorithm along with its recommended settings, suit well infrastructure managed networks which are permanent and last for several days, weeks and months while it does not suit pop-up Wi-Fi Direct networks running for short periods of time on devices with limited processing capabilities and power constraints. Rearticulating, there is no mechanism in Aruba's algorithm to stop the network from moving between channels when the threshold is hit. The stability of the network is only assured by adopting a time-tolerant settings which suit the lifetime of an infrastructure managed network and not the lifetime of a pop-up Wi-Fi Direct network. As was also shown in the last experiment, the proposed algorithm outperforms the algorithm of [32] when both of them are employed in Wi-Fi Direct networks.

## References

1. IEEE 802.11 Standard for Information Technology. (2012). Telecommunications and information exchange between systems local and metropolitan area networks—Specific requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications (pp. 1–2793).
2. Wi-Fi Alliance. (2014). Wi-Fi Peer-to-Peer (P2P) Technical Specification v1.5, Wi-Fi Alliance Specification (pp. 1–183).
3. Bluetooth Special Interest Group. (2010). Bluetooth Specification V.4 (pp. 1–2302).
4. Wannstrom J. (2013). *LTE-Advanced for 3GPP*. <http://www.3gpp.org/technologies/keywords-acronyms/97-lte-advanced>. Accessed January 21, 2016.
5. GPP TR 22.803. (2013). Feasibility study for Proximity Services (ProSe). v.12.2.0 (pp. 1–45).
6. AlWreikat, L., Chai, R., & Abu-Sharkh, O. M. F. (2014). Energy-efficiency based resource allocation for D2D communication and cellular networks. In *4th IEEE BdCloud*, Sydney, NSW (pp. 195–200).
7. Lin, X., et al. (2014). An overview of 3GPP device-to-device proximity services. *IEEE Communications Magazine*, 52(4), 40–48.

8. Wu, Y., Guo, W., Yuan, H., Li, L., Wang, S., Chu, X., & Zhang, J. (2016). Device-to-device (D2D) meets LTE-Unlicensed. *IEEE Communications Magazine* (in press).
9. Asadi, A., & Mancuso, V. (2013). *WiFi Direct and LTE D2D in action* (pp. 1–8). Valencia: IFIP Wireless Days.
10. Asadi, A., Wang, Q., & Mancuso, V. (2014). A survey on device-to-device communication in cellular networks. *IEEE Communications Surveys & Tutorials*, 16(4), 1801–1819.
11. Ghavimi, F., & Chen, H. H. (2015). M2M communications in 3GPP LTE/LTE-A networks: Architectures, service requirements, challenges, and applications. *IEEE Communications Surveys & Tutorials*, 17(2), 525–549.
12. Gu, J., et al. (2015). A resource allocation scheme for device-to-device communications using LTE-A uplink resources. *Pervasive and Mobile Computing*, 18, 104–117.
13. IEEE Std 802.11n. (2009). Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: Enhancements for higher throughput (pp. 1–536).
14. IEEE Std 802.11ac. (2013). Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: Enhancements for very high throughput for operation in bands below 6 GHz (pp. 1–425).
15. IEEE Std 802.11ad. (2012). Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: Enhancements for very high throughput in the 60 GHz Band (pp. 1–628).
16. IEEE802.11 High Efficiency WLAN Study Group. Status of Project IEEE 802.11ax. [http://www.ieee802.org/11/Reports/tgax\\_update.htm](http://www.ieee802.org/11/Reports/tgax_update.htm). Accessed October 15, 2015.
17. Deng, D., et al. (2014). IEEE 802.11ax: Next generation wireless local area networks. In *The 10th international conference on heterogeneous networking for quality, reliability, security and robustness*, Rhodes (pp. 77–82).
18. Bellalta, B. (2015). IEEE 802.11ax: High-efficiency WLANs. arXiv preprint [arXiv:1501.01496](https://arxiv.org/abs/1501.01496).
19. Zhang, J., Wang, J. M., & Bensaou, B. (2013). Optimal resource sharing in multi-cell wireless LANs: Joint channel assignment, association control, airtime sharing and contention resolution. In *The 9th IEEE IWCMC*, Cagliari, Sardinia, Italy (pp. 317–322).
20. Kauffmann, B., Baccelli, F., & Chaintreau, A. (2007). Measurement-based self organization of interfering 802.11 wireless access networks. In *IEEE INFOCOM*, Anchorage, AK (pp. 1451–1459).
21. Xu, W., Hua, C., & Huang, A. (2011). Channel assignment and user association game in Dense 802.11 wireless networks. In *IEEE ICC* (pp. 1–5).
22. Wang, Y., et al. (2014). Coordinated fair resource sharing in dense indoor wireless networks. In *IFIP networking conference*, Trondheim (pp. 1–9).
23. Karimi, O. B., Liu, J., Rexford, J. (2014). Optimal collaborative access point association in wireless networks. In *IEEE INFOCOM*, Toronto (pp. 1141–1149).
24. Ozyagci, A., Sung, K. W., & Zander, J. (2014). Association and deployment considerations in dense wireless LANs. In *The 79th IEEE vehicular technology conference*, Seoul (pp. 1–5).
25. Gong, D., et al. (2012). AP Association in 802.11n WLANs with heterogeneous clients. In *IEEE INFOCOM*, Orlando, FL (pp. 1440–1448).
26. Gong, D., & Yang, Y. (2014). On-line AP association algorithms for 802.11n WLANs with heterogeneous clients. *IEEE Transactions on Computers*, 63(11), 2772–2786.
27. Ji, B., Song, K., Zhu, J., & Li, W. (2014). Efficient MAC protocol design and performance analysis for dense WLANs. *Wireless Networks*, 20(8), 2237–2254.
28. Yue, X., Wong, C.F., & Chan, S. H. G. (2010). A distributed channel assignment algorithm for uncoordinated WLANs. In *IEEE CCNC*, Las Vegas (pp. 1–5).
29. Chaves, F. S., et al. (2014). Adaptive transmit power for Wi-Fi dense deployments. In *The 80th IEEE vehicular technology conference*, Vancouver, BC (pp. 1–6).
30. Jiang, H., et al. (2015). TDOCP: A two-dimensional optimization integrating channel assignment and power control for large-scale WLANs with dense users. *Ad Hoc Networks*, 26, 114–127.
31. Jamil, I., Cariou, L., & Helard, J. F. (2014). Improving the capacity of future IEEE 802.11 high efficiency WLANs. In *The 21st international conference on telecommunications*, Lisbon (pp. 303–307).
32. Abu-Sharkh, O., Qandah, R., Baba, M., & Sabah, M. (2012). Dynamic frequency allocation scheme for a standalone wireless access point. In *The 35th IEEE Sarnoff Symposium*, Newark, NJ (pp. 1–5).
33. Aruba Networks. *Adaptive radio management*. [http://www.arubanetworks.com/pdf/solutions/TB\\_ARM.pdf](http://www.arubanetworks.com/pdf/solutions/TB_ARM.pdf). Accessed January 21, 2016.
34. Sagari, S., et al. (2013). Performance evaluation of mobile hot-spots in densely deployed WLAN environments. In *The 24th IEEE International Symposium on Personal Indoor and Mobile Radio Communications*, London (pp. 2935–2939).
35. Cardoso, K. V., & de Rezende, J. F. (2012). Increasing throughput in dense 802.11 networks by automatic rate adaptation improvement. *Wireless Networks*, 18(1), 95–112.
36. Santos, M. A., Villalón, J. M., Orozco-Barbosa, L. (2012). Rate adaptation algorithm for 802.11 networks: A dynamic decision approach. In *The 5th joint IFIP wireless and mobile networking conference*, Bratislava (pp. 75–80).
37. Santos, M. A., et al. (2016). Dyn-ARF: A rate adaptation mechanism sensitive to the network load over 802.11 WLANs. *Telecommunication Systems*, 61(1), 5–19.
38. Heusse, M., et al. (2003). Performance anomaly of 802.11b. In *IEEE INFOCOM*, San Francisco (pp. 836–843).
39. Abinader, F. M., et al. (2014). Performance evaluation of IEEE 802.11n WLAN in dense deployment scenarios. In *The 80th IEEE vehicular technology conference*, Vancouver, BC (pp. 1–5).
40. The Network Simulator NS-2. <http://www.isi.edu/nsnam/ns/>. Accessed October 15, 2015.
41. Camps-Mur, D., Garcia-Saavedra, A., & Serrano, P. (2013). Device-to-device communications with Wi-Fi Direct: Overview and experimentation. *IEEE Wireless Communications*, 20(3), 1536–1584.
42. Farproc WIFI Analyzer. <http://a.farproc.com/wifi-analyzer>. Accessed October 15, 2015.
43. GNU General Public License version 2. <http://www.gnu.org/licenses/gpl-2.0.html>. Accessed October 15, 2015.



**Osama M.F. Abu-Sharkh** received the B.Sc. degree from University of Jordan, Amman, Jordan, in 1999, the M.Sc. degree from University of North Carolina, Charlotte, NC, USA, in 2001 and the Ph.D. degree from University of Minnesota Twin Cities, Minneapolis, MN, USA, in 2006, all in electrical engineering. He joined the Communications Engineering Department at Princess Sumaya University for Technology, Amman, Jordan in September 2007. He served as the Chair of the Communications Engineering Department from February 2012 to September 2014. He is currently with the Computer Engineering Department, Networks and

Information Security Engineering program. He is a member of the IEEE communications, computer, vehicular technology, signal processing and intelligent transportation systems societies. He is also a member of ACM and a fellow member of the applied innovation institution. He participated in organizing and conducting many local and international conferences and events. His current research interest include wireless telecommunications and computer networks, smart cities, wireless vehicular networks, smart grids, Quality of Service, multimedia communications and processing, optical communications, wireless sensor networks, wireless positioning systems and resource allocation and management.



**Esam A. AlQaralleh** received the B.Sc. degree from University of Jordan, Amman, Jordan, in 1995, the M.Sc. and Ph.D. degree from National Chiao Tung University, Taiwan, in 2006, all in electrical engineering. He joined the Computer Engineering Department at Princess Sumaya University for Technology, Amman, Jordan in Sep 2006. He served as the Chair of the Computer Engineering Department from 2009 to 2011. He is a member of the

IEEE computer, and RAs societies. He participated in organizing and conducting many local and international conferences and events. His current research interest include video coding/motion estimation: algorithm and implementation using FPGA, Embedded Systems/RTOS, Engineering Education and Arabic Handwritten Recognition.



**Omar M. Hasan** received his B.Sc. degree in electrical engineering, from New Mexico State University, USA, in 1987, M.Sc. and Ph.D. degrees in communication engineering, from New Mexico State University, USA, in 1990 and 1996 respectively. He joined Princess Sumaya University for Technology in 1997 where he served as chairman for the communications engineering department in the period 2005–2010 and dean of student

affairs. During 1991–1995, Dr. Hasan worked as system manager at Buchanan & associates, El Paso, Texas, USA. He worked in different projects at physical science laboratory, Las cruces, New Mexico, USA, in the period 1991–1992. In summer 2000, he worked at Global Cardiac Monitors, research and development division in the area of Turbo coding, Webster, Texas, USA. He worked as a system analyst at Intelligent Technology Incorporated, communication division, Las Cruces, New Mexico, USA 1989–1992. He worked as a consultant for Global Cardiac Monitoring, Webster, Texas, USA, 2001, 2004, and 2010–2011, 2014. He worked as a research assistant at the manual Lujan, Jr. center for Space and Telemetry and Telecommunications, Simulation and Tracking system division, New Mexico State University, Las Cruces, New Mexico, USA, 1996.