IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 30, 2019, accepted February 24, 2019, date of publication March 5, 2019, date of current version March 25, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2903228

A Group-Less and Energy Efficient Communication Scheme Based on Wi-Fi Direct Technology for Emergency Scenes

MENG-SHIUAN PAN AND CHUNG-MING WANG
Department of Computer Science and Information Engineering, Tamkang University, New Taipei 251, Taiwan
Corresponding author: Meng-Shiuan Pan (mspan@mail.tku.edu.tw)

This work was supported by the MOST under Grant 107-2221-E-032-017.

ABSTRACT Recently, smartphones are commonly equipped with Wi-Fi direct modules, which can be used to support infrastructure-less communications. Considering an emergency scene, communication infrastructures are damaged or unable to operate. Affected people in the emergency site cannot communicate with each other and with the outside world until rescue personnel establishes an emergency network for them. In this paper, we aim to provide an emergency communication method based on Wi-Fi direct technology. Originally, Wi-Fi direct capable devices rely on a Wi-Fi direct group mechanism to operate. However, we observe that forming Wi-Fi direct groups may not be appropriated in the emergency scenario. Besides, Wi-Fi direct capable devices carried by affected people should operate in power saving mode when emergency. In this paper, we model the target scenario as an emergency Wi-Fi direct device scheduling (EWDS) problem. We then propose centralized and distributed schemes to solve the EWDS problem. The proposed schemes schedule devices' active and sleep timings with the guarantee that all devices can join the network within a bounded time and devices can exchanges packets with each other effectively. Furthermore, by the proposed distributed schemes, devices can operate locally and unsynchronized without a central controller. We evaluate the proposed schemes by simulations and prototyping implementation, and the results demonstrate the effectiveness of our designs.

INDEX TERMS Emergency scenes, energy efficient, group-less communications, sleep scheduling, smartphone, Wi-Fi direct.

I. INTRODUCTION

Following the widespread adoption of smartphone technology, people have become reliant on cellular networks and Wi-Fi wireless networks for communication. These two networks require infrastructures (i.e., cellular base stations and Wi-Fi access points (APs)) to serve. However, when emergency incidents occur, infrastructures may be damaged or unable to operate. (For example, in 2010 Haiti earthquake, the public telephone system was unavailable for more than one week. [4]) If there are affected people in the emergency location, they cannot communicate with each other and with outside world, and thus increases rescue efforts. So, a communication method without relying on infrastructures for affected people when emergency is needed [8], [19], [23].

Most modern smartphones are equipped with Bluetooth and Wi-Fi direct modules. Both of these two wireless modules can operate without infrastructures. As analyzed in [22], the Wi-Fi direct module can support a longer communication range (up to 200 meters) and have better communication performances than the Bluetooth module. Thus, Wi-Fi direct technology is more suitable for supporting emergency services. Moreover, when emergency, affected people may not be able to charge their smartphones' batteries, and they may need to wait for rescue service in several hours. So, smartphones have to stay in power saving mode to wait for rescue personnel starting an emergency network. After the emergency network is formed, smarphones still need to operate in low duty cycle mode until the affected people are safe.

Given a set of devices with Wi-Fi direct capability, Fig. 1 shows a Wi-Fi direct network. In the network, there is a *group owner* (*GO*). A device can join the network as a client device

The associate editor coordinating the review of this manuscript and approving it for publication was Byung-Seo Kim.

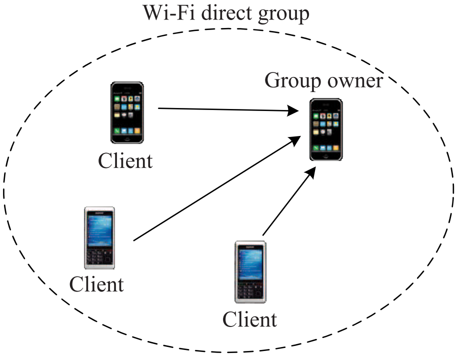

FIGURE 1. **A Wi-Fi direct network.**

after connecting to the GO. Like a Wi-Fi AP, the GO is responsible for handling packet exchanges between devices in the group. According to the Wi-Fi direct specification [5], devices that are going to join the network have to perform a *device discovery procedure* in advance. During the procedure, devices can obtain the GO's information by exchanging *probe request packets* and *probe response packets* with the GO. Then, devices can connect to the GO by an *association procedure*. We observe that the device discovery and association procedures have the following two issues when applying to the emergency scenario. First, these two procedures need manual operations by users (e.g., selecting a device to connect from a list of devices or entering PIN to join the group). Those manual operations are inconvenient to rescuers and affected people. Second, the discovery and association procedures are time consuming. We conduct a preliminary experiment on a Wi-Fi direct network, which contains one GO and four client devices. In the experiment, those four client devices form a rectangular and the GO is placed at the center. The distances between the GO and client devices are 3 meters, and there is no obstacle between GO and client devices. The GO first starts the network, and these four client devices start to search the GO at the same time. We measure the time needed for these four devices to connect to the GO, and the experiment results reveal that needed times are 86.5 to 275 seconds (excluding manual operations of users). So, based on the above two observations, a method that can quickly and easily launch Wi-Fi direct communication service is needed. Besides, according to the specification, a Wi-Fi direct group is a single-hop network. When emergency, affected people may be distributed in a large area. Thus, a design that can support multi-hop transmissions in Wi-Fi direct networks is also needed.

In this work, we propose a group-less and energy efficient communication scheme based Wi-Fi direct technology for emergency scenes. When emergency happens, smartphones hold by affected people (say '*device*' in short) first perform sleep and wake up schedules to save battery power. Once rescue personnel arrive in the emergency site, a *rescue device* establishes an emergency network for affected people. The emergency network will contain no Wi-Fi direct group, and devices can exchange messages (from affected people or rescue personnel) through the modified probe request and probe response packets directly. Messages from and to affected people can be disseminated through multi-hop transmissions. When establishing the emergency network, we further consider two goals. First, all devices carried by affected people should be able to join the emergency network. Second, the needed time for rescue device to know the existence of all devices (or say *network discovery latency*) should be minimized. We model the above two goals by an *emergency Wi-Fi direct device scheduling* (*EWDS*) problem. Then, we propose a centralized solution and a distributed solution to solve the EWDS problem. In the centralized solution, devices are considered to be synchronized and there is a centralized controller in the network. Under some network conditions, the network discovery latency of the centralized solution can be optimized. In order to relieve the restriction all devices should be synchronized, the distributed solution utilizes the *grid-quorum* concept [11] to arrange devices' schedules. Although there is no centralized controller in the network, by the distributed solution, all devices can still be discovered and the network discovery latency can be bounded. The simulation results indicate that the proposed schemes can effectively shorten the time needed for discovering devices. Besides, we modify the Android source codes to implement the designed distributed scheme. We further implement acknowledgement (ACK) and negative acknowledgement (NACK) mechanism to ensure reliable transmissions between devices. The experiment and implementation results indicate that the designed scheme can indeed achieve quick and reliable communications, and can also reduce devices' power consumptions.

The main contribution of this paper is that we propose a concept of establishing group-less and energy efficient Wi-Fi direct networks for emergency services, and the proposed schemes have the following features:

1) The proposed schemes get rid of the restriction on establishing one-hop Wi-Fi direct groups, and the formed group-less emergency network can support multi-hop transmissions. Besides, devices can attach to the emergency network quickly and easily.
2) In the proposed schemes, devices operate in low duty cycle mode when emergency. Although devices are almost stayed in sleep mode, the designed scheme can guarantee that devices will be discovered within a bounded time.
3) In the proposed distributed scheme, time between devices can be unsynchronized, and devices can operate without a central controller. This feature is practical and essential for the needs when emergency.
4) The designed schemes can be compatible with the original Wi-Fi direct protocol.

Besides, based on our survey, this is the first work that dedicates to utilize Wi-Fi direct technology for emergency communications and rescue services.

The rest of this paper is organized as follows. Section II overviews the Wi-Fi direct protocol. Section III reviews some

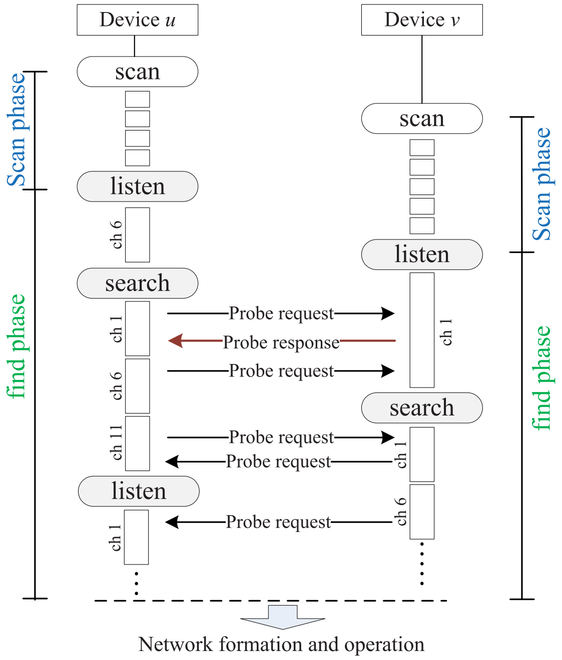

FIGURE 2. Wi-Fi direct discovery procedure.

previous works. Section IV and Section V present the network model and the proposed schemes, respectively. Then, simulation and implementation results are shown in Section VI and Section VII, respectively. Finally, Section VIII concludes this paper.

II. OVERVIEW ON Wi-Fi DIRECT

The goal of Wi-Fi direct is to allow Wi-Fi capable devices to be able to communicate with each other through direct links. According to the specification [5], when a device enters a network, this device needs to perform *device discovery procedure* to find a group owner (GO). As shown in Fig. 2, the device discovery procedure can be divided into a *scan phase* and a *find phase*. Assume that a device D_v does not join a group. When D_v starts the discovery procedure, it first performs the scan phase to collect information of its nearby devices or existing groups in all Wi-Fi direct *social channels* (i.e., channel 1, 6, and 11 in the 2.4 GHz band). After the scan phase, D_v enters the *find phase*. In the find phase, D_v alternatively switches between *listen state* and *search state*. In the beginning, D_v stays in the listen state in a channel for a time interval with length 100, 200, or 300 time units (where the length of a time unit is defined in IEEE 802.11 specification [1]). If D_v receives a *probe request packet* from another device, say D_u, device D_v will then reply a *probe response packet* to D_u. After exchanging the probe request and response packets, devices D_v and D_u are said to have been discovered by each other. Then, D_v can enter search state and then broadcasts its probe request packets to discover other devices in Wi-Fi direct social channels. According to the specification, device D_v can associate to the group that it learns in the find phase. If there is no existing group found in the find phase, D_v can complete with other devices (using the *group owner negotiation procedure* in [5]) for becoming a group owner.

In this work, we propose to establish a Wi-Fi direct emergency network without forming Wi-Fi direct groups. Therefore, devices do not need to collect information of existing groups, and thus the scan phase can be ignored. In our scheme, devices only operate in a fixed social channel, and they stay in the modified find phase to perform listen state operation, search state operation, or entering sleep mode. During the search state, devices are configured to continuously broadcast probe request packets. In the listen state, devices wait for probe request packets from nearby devices. It is not hard to see that devices operating in search state will consume more energy. To avoid unnecessary energy consumption, the proposed schemes also schedule devices to switch between listen and search states adaptively.

III. RELATED WORKS

We first review some schemes in references [7], [12], [15], [21], [24], and [25], which aim to establish communication networks in emergency scenes. Chatzimilioudis *et al.* [12] consider the scenario that cellular networks become non-operational when emergency. At this time, an overlay network based on the k-nearest-neighbor (kNN) concept is established to offer communication services. However, the proposed solution is a centralized algorithm, which may not be suitable for emergency scenarios. Aloi *et al.* [7] propose a distributed scheme to form emergency networks. In the proposed scheme, each device locally decides its role, as a router or the gateway, according to its remaining energy. Then, devices select routing path locally by themselves based on swarm intelligence concept. However, the proposed scheme in [7] needs a lot of time to converge and cannot be applied to Wi-Fi direct capable devices. Reference [25] introduces a scheme that borrows radio spectrums from commercial users to support quality of service (QoS) data transmissions when emergency. But, the proposed scheme relies on a central controller, i.e., base station, to operate. Moreover, reference [21] introduces a multicast scheme for public safety based on Wi-Fi direct networks. Before multicasting, devices select some channels that can provide better quality, and then report to their group owner. The group owner then decides multicast trees by considering overall system performance. The proposed scheme can increase throughput of multicasting, but it can only be used in single hop networks. Kuada and Bannerman [24] propose to utilize Wi-Fi direct module to form an opportunistic rescue network. However, the proposed scheme only allows single hop transmissions. Furutani *et al.* [15] propose a scheme to achieve efficient information diffusion through multiple hops in an emergency network. In the proposed scheme, devices are grouped into virtual cells, and they can dynamically form and dismiss their groups to facilitate packet exchanges. However, the proposed scheme is applicable under the condition that devices are time synchronized.

Next, we introduce some previous works that discuss data transmissions and group formation methods for Wi-Fi direct networks. First, references [9], [10], [14], [18], [20], and [34] propose schemes for inter-group communications in Wi-Fi direct networks. Reference [20] utilizes the concurrent mode specified in the Wi-Fi direct specification to achieve inter-group communication. Assume that there are two Wi-Fi direct groups. In the designed scheme, there will have a bridge device, which has two radio interfaces. The bridge node can simultaneously be one group's owner and another group's client, and is responsible for relaying packet among these two groups. Reference [10] presents a scheme to use one radio interface to achieve concurrent mode operation. In the proposed scheme, a bridge node first forms a group and accepts connections from other devices. The bridge node also performs legacy Wi-Fi association procedure to connect to another group owner. After forming groups, bridge devices use Wi-Fi broadcast mechanism to send packets to its group clients. Reference [14] introduces a solution to achieve multi-hop data disseminations between Wi-Fi direct capable devices. In the proposed scheme, devices will switch their roles between as legacy clients or group members/owners to facilitate packet exchanges. Reference [18] introduces a policy to select group owners in Wi-Fi direct networks. Given a set of devices, the proposed scheme selects a group owner that has better radio signal qualities among its neighbors. Yao *et al.* [34] propose a group communication scheme based on delay-tolerant routing concept. When a device wants to send a packet to another device, this device first sends the packet to its group owner. Then, the group owner checks its routing table. If the destination device does not locate in its group, the group owner finds several devices, which have lower received signal strength indication (RSSI) values, as the next hops. Reference [9] introduces a middleware for maintaining connections between Wi-Fi capable devices autonomously. Following the Wi-Fi direct specification, each device first joins a Wi-Fi direct group. The middleware selects some devices in the formed groups to be bridge nodes, which are used to facilitate relaying packets among groups. Devices can then initiate unicast or multicast transmissions among the established network topology. In summary, the designs in [9], [10], [14], [18], [20], and [34] are more suitable for general data transmissions. Wi-Fi direct groups are still needed in these schemes. Besides, those schemes do not consider the power saving issue. Moreover, Lee *et al.* [26] propose a concept of using low-power Bluetooth device discovery procedure to replace the original Wi-Fi direct device discovery procedure. The reason is that the Bluetooth device discovery procedure can perform faster. But, this design cannot be compatible with the Wi-Fi direct specification. Furthermore, references [29] and [35] propose to attach messages by modifying service discovery request frames and probe request frames in the Wi-Fi direct protocol to provide communication services. However, the proposed schemes in [29] and [35] cannot guarantee successful delivery and do not consider power saving. Reference [16] designs data dissemination policies in Wi-Fi based peer-to-peer networks. When disseminating a packet to network devices, a central server takes into account the battery capacities and bandwidth of devices, and then decides downlink paths for the packet. Although the proposed scheme can effectively lengthen devices' lifetime, the proposed scheme cannot be used in the emergency scenes because that it is a centralized scheme.

References [5], [13], and [31]–[33] propose power saving schemes for Wi-Fi and Wi-Fi direct modules. Usman *et al.* [13] observe that when using high bandwidth Wi-Fi modules to transmit packet, there are many time gaps between individual packets. So, during these gaps, the Wi-Fi module can go to doze mode to conserve devices' battery power. Reference [32] introduces a scheme to conserve devices' power by allowing Wi-Fi modules to scale down sampling rates. Although the sampling rates are decreased, the proposed scheme can still guarantee decoding performance and the reliability of communications. Although both [13] and [32] can indeed conserve devices' power, the proposed schemes cannot be applied on Wi-Fi direct modules directly. Moreover, the Wi-Fi direct specification [5] defines two energy saving schemes, which operate based on Wi-Fi direct groups. In the first scheme, group clients locally decide whether to go to sleep mode. When the group owner realizes all its clients are in sleep mode, it can also enter sleep mode. In the second one, a group owner announces that it is going to enter sleep mode, and then all its clients follow the group owner's schedule to sleep. Based on the defined two schemes in the Wi-Fi direct specification, reference [33] further introduces a mixed scheme to switch these two energy saving schemes adaptively. Usman *et al.* [31] introduce a policy to conserve energy by controlling the Wi-Fi direct group size and coordinating devices' transmission power. Although the proposed schemes in [5], [32], and [33] are designed for Wi-Fi direct module, these schemes rely on Wi-Fi direct group mechanism to operate.

IV. NETWORK MODEL

Given an emergency site, there are affected people carrying smartphones with Wi-Fi direct capability (say *devices* in short). In this work, we model a graph $G = (\overline{V}, \overline{E})$, where $\overline{V}$ contains those devices carried by affected people and the *rescue device* D_R carried by rescue personnel, and $\overline{E}$ contains all symmetric radio links between devices in $\overline{V}$. When emergency, the rescue device can disseminate messages to affected people, and devices carried by affected people can report messages to rescue personnel. To facilitate data transmissions, the rescue device D_R forms a tree network rooted at it, and the tree (i.e., the emergency network) is started by D_R broadcasting its probe request packets. A device, say D_v, can join the network by setting a probe request sender as its *parent device*. Then, D_v can reply a probe response packet carrying messages from the affected person to its parent. Device D_v can also help to expand the network by sending its probe request packets and relay messages from its descendant

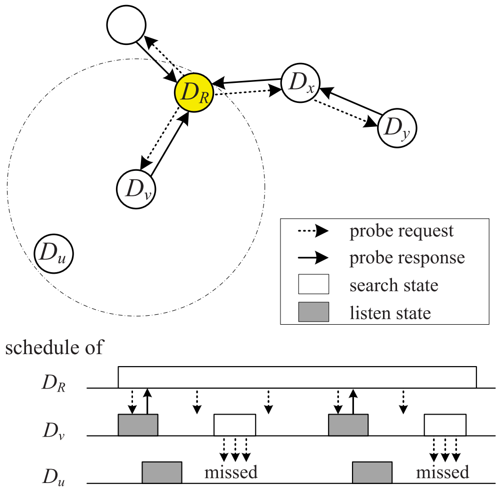

FIGURE 3. **The network model for emergency communication.**

devices to the D_R. Fig. 3 shows an example of a formed tree network.

We assume that the D_R is powered by a battery with large capacity or can recharge its battery easily, and thus D_R always keeps in search state to broadcast its probe request packets continuously. Moreover, those devices (carried by affected people) will switch between listen and search states or go to sleep (as shown in Fig. 3) to prolong their lifetime. Besides, in the emergency network, devices are configured according to the following two duty cycle requirements.

- If a device does not decide a parent device, it only executes listen state operations (as the device D_u in Fig. 3). This device's active time should be less than a duty cycle requirement C_L.
- If a device has decided a parent device, it can execute its search state operations. This device also wakes up to stay in listen state for a while to track its parent device's search state (as the device D_v in Fig. 3). The device's active time for executing search state should be less than a duty cycle requirement C_S.

In this work, we schedule devices' listen and search states to facilitate emergency communications. When scheduling, we have the following two considerations. First, we have to guarantee that each device can find a probe request packet sender as its parent device. Assume that a device D_v has decided a parent device, and D_v can start to broadcast its probe request packets. It is possible that a device D_u, which is located in the transmission range of D_v, cannot receive D_v's probe request packets because that D_u's listen state does not align to D_v's search state (as shown in Fig. 3).

Definition 1: For a device $D_v \in \{\overline{V} \setminus D_R\}$, we say that D_v is a *orphan* device if D_v's listen state "cannot" meet any devices' search state.

The second consideration is that we have to decrease the latency of discovering all devices in the network. In this work, a device is said to be discovered if the D_R receives its report. So, device D_v's *discovery latency* $d_{dis}(D_v)$ is defined as the interval from the time that D_R broadcasts its first probe request packet to the time that D_R receives the first report from D_v. For example, assume that D_R broadcasts its first probe request packet at time t_1, and then D_R receives the first report from D_v at time t_2. The discovery latency of D_v will be $L_{dis}(D_v) = (t_2 - t_1)$. Then, we can define the network discovery latency.

Definition 2: The *network discovery latency* $L_D = \max\{L_{dis}(D_v)\}, \forall D_v \in \{\overline{V} \setminus D_R\}$.

Then, we formally define the objective of deciding devices' schedules.

Definition 3: Given an emergency network $G = (\overline{V}, \overline{E})$, the duty cycle requirements C_L and C_S, the emergency Wi-Fi direct device scheduling (EWDS) problem is to decide a schedule of devices' listen and search states such that (i) there is no orphan in $\overline{V}$ and (ii) L_D is minimized.

Note that the EWDS problem can be taken as a slot assignment problem. In the literature, there are many slot assignment solutions (e.g., [17], [28]) for wireless sensor networks. In those previous works, beacons are used to identify the starts of time slots, and thus the transmissions of beacons should be interference-free. So, the assignment of slots should avoid interferences between devices, i.e., devices that are located within two hops cannot be assigned to the same slot. Unlike those previous works, in this work, nearby devices can start their search states or listen states at the same time. The rationale is that in the lower layer of Wi-Fi direct module, the nature of repeated probe request packet transmissions in the search state facilitate successful delivery on messages. Besides, the size of messages in the probe request and response packets are limited so the network traffic loads will not be heavy. Thus, the needs on dedicate and interference-free time periods for search and listen states can be omitted.

V. THE PROPOSED SOLUTIONS FOR EWDS PROBLEM

In this section, we introduce two solutions to solve the EWDS problem.

A. THE CENTRALIZED SOLUTION

By the settings of duty cycle requirements C_L and C_S, we can decide a variable $C' = \min\{C_L, C_S\}$, where C' is a duty cycle value that satisfies both of the duty cycle requirements. In the centralized scheme, network time is divided into repeated and continuous time frames as shown in Fig. 4. The size of a time frame is T_F unit of time. To facilitate discovering devices and relaying messages, a device will execute at most twice search states and at most twice listen states in a time frame. Then, by T_F and C', the time duration for one search or listen state operation, say t_o, can be obtained by the following Eq. (1).

$$t_o = \frac{T_F \times C'}{4} \qquad (1)$$

Note that the number "four" in the denominator in Eq. (1) represents the summation of two search states and two listen states in a time frame.

Before arranging schedules of devices' search and listen states, we first form a breadth-first search (BFS) tree $\overline{T}$ rooted at D_R in G. By $\overline{T}$, we can obtain the following two parameters: (i) The depth of each device $D_v \in \overline{V}$, say $dep(D_v)$, on $\overline{T}$. (ii) The maximum depth M^T_{dep} of $\overline{T}$.

Let t_{start} represent the time instant of a time frame. In this scheme, devices follow the same schedule in every time frame. Without loss of generality, we only discuss devices' schedules of the time frame started from t_{start}. Given a device D_v, its schedule is decided by the following rules.

1) If D_v locates in depths $1..(M^T_{dep} - 2)$, D_v will execute two search states and two listen states in the time frame. The schedule of D_v is decided by the following rules.
   - D_v's first listen state starts at time instant $t^l_1 = t_{start} + t_o \times (dep(D_v) - 1)$.
   - D_v's first search state starts at time instant $t^s_1 = t^l_1 + t_o$.
   - D_v's second search state starts at time instant $t^s_2 = t^s_1 + t_o \times (M_{dep} - dep(D_v) - 2) \times 2 + 1)$.
   - D_v's second listen state starts at time instant $t^l_2 = t^s_2 + t_o$.
2) If D_v locates in depth $M^T_{dep} - 1$, D_v will execute one search state and two listen states in the time frame. The D_v's schedule is decided by the following rules.
   - D_v's first listen state starts at time instant $t^l_1 = t_{start} + t_o \times (dep(D_v) - 1)$.
   - D_v's first search state starts at time instant $t^s_1 = t^l_1 + t_o$.
   - D_v's second listen state starts at time instant $t^l_2 = t^s_1 + t_o$.
3) If D_v locates in depth M^T_{dep}, D_v executes only one listen state at time instant $t^l_1 = t_{start} + t_o \times (dep(D_v) - 1)$.

Fig. 4 shows an example of the above schedule, where $M^T_{dep} = 5$. By the above rules, the arrangement of search and listen states looks like a 'V' shape. The left portion of the 'V' is to facilitate (i) discovering devices when the network starts and (ii) data dissemination from the D_R after the network is formed. On the other hand, the right portion of the 'V' is used to facilitate data reporting from devices to D_R.

Theorem 1: Given the tree $\overline{T}$ and t_o, the network discovery latency L_D is optimal in $O((2 \times M^T_{dep} - 1) \times t_o)$.

Proof: The network discovery latency is dominated by the discovery time of devices that located in depth M^T_{dep}. Let S_d represent the set of devices that locate in depth d. Assume that the D_R starts broadcasting probe request packet at time instant t. In our design, devices in S_1 can be discovered within $t + t_o$. After devices in S_1 start their search state at time $t + t_o$, devices in S_2 can receive probe request packets within $t + 2t_o$. But, the messages (carried in probe response packets) generated by S_2 cannot reach D_R directly. So, devices in S_2 will be discovered by D_R until devices in S_1 enter listen state again and relay messages from S_2 to D_R.

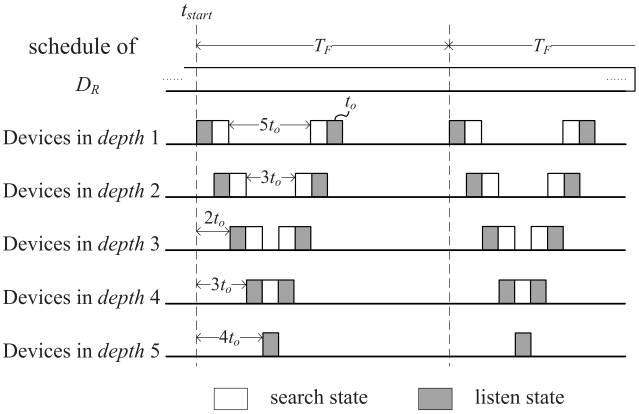

FIGURE 4. An example of the centralized scheme.

Following the similar flow, messages from devices in $S_{M^T_{dep}}$ can be sent to devices in $S_{M^T_{dep}-1}$ in the second search states of devices in $S_{M^T_{dep}-1}$. Then, messages can be relayed to D_R through ancestor devices by the staggered search and listen state schedule with no additional latency. So, the proposed scheduling method can achieve optimal network discovery latency under the given $\overline{T}$ and t_o,

By the definition, network discovery latency is the interval between the first and last listen state of the devices in S_1. The time interval is the summation of (i) needed slots of finding all devices, i.e., the left 'V' part, $t_o \times (M^T_{dep} - 1)$, (ii) needed slots for the devices in $S_{M^T_{dep}}$ to report their messages, i.e., the right 'V' part, $t_o \times (M^T_{dep} - 1)$, and (iii) one listen state period, i.e., one t_o, for devices in $S_{M^T_{dep}}$. As a result, the network discovery latency will be $O((2 \times M^T_{dep} - 1) \times t_o)$. ☐

B. THE DISTRIBUTED GRID-QUORUM SOLUTION

The above centralized solution may not be applicable in real cases because of the following two reasons. First, in the centralized solution, devices have to be synchronized. In the literature, many existing synchronization schemes are proposed for ad hoc or sensor networks [27], [30], and these schemes demand devices to exchange control messages to preserve time synchronization. But, according to the Wi-Fi direct specification, devices cannot exchange packet arbitrarily,[1] and thus existing synchronization schemes cannot be applied. So, we claim that the designed scheme should allow devices to operate locally (without time synchronization). Second, in most emergency scenes, the transmission range of the central device cannot cover all devices. As a result, devices that are located more than 1-hop away from the central device cannot know the exact start timing of D_R's search state operation because of latencies on relaying. So, we claim that the designed scheme should be a decentralized one.

According to the above discussions, the designed scheme should be decentralized and devices are unsynchronized.

---

[1] Recall that in the original specification, devices cannot exchange packet with each other unless they are in the same group.

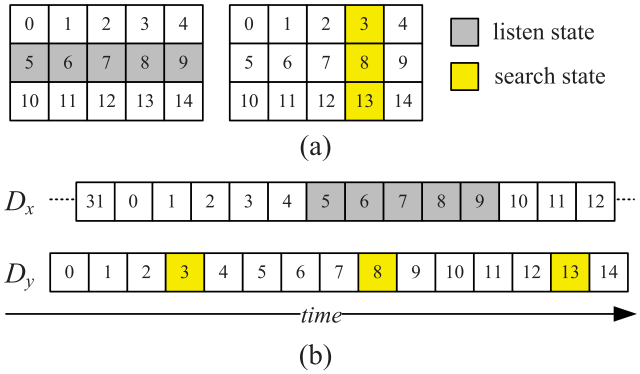

FIGURE 5. An example of the listen state and search state schedules of devices D_x and D_y.

So, it is a challenge that how to guarantee there is no orphan device in the network. In other words, when scheduling, we have to ensure that each device's listen state is able to overlap with another devices' search states. To achieve this goal, we utilize the concept of *grid-quorum* [11] to design the distributed scheme.

In the proposed scheme, the network time is divided into *slots*, and the size of a slot is t_s. Every $q = (q_m \times q_n)$ consecutive slots are grouped together and called a time frame. Those slots in a time frame can logically be arranged as a grid with q_m rows and q_n columns. For example, Fig. 5(a) shows two grid quorums, and each of them is with size $3 \times 5 = 15$ slots. In a time frame, each slot is labeled as slot numbered s_i, where $i \in \{0 .. (q_m \times q_n - 1)\}$.

Given the duty cycle requirements C_L and C_S, this scheme decides the values of q_m and q_n by the following two equations.

$$q_m = \lceil \frac{1}{C_L} \rceil \tag{2}$$

$$q_n = \lceil \frac{1}{C_S} \rceil \tag{3}$$

Then, a device $D_v \in \overline{V}$ decides its schedule of listen and search states by following rule.

- Rule 1: If D_v does not decide a parent device, D_v randomly decides a value $k \in \{0 .. (q_m - 1)\}$. Then, D_v executes its listen state in slots numbered $s_{(q_n \times k)}, s_{(q_n \times k+1)}, .., s_{(q_n \times k+q_n-1)}$.
- Rule 2: If D_v has decided a parent device, D_v randomly decides a value $l \in \{0 .. (q_n - 1)\}$. Then, D_v executes its search state in slots numbered $s_l, s_{(q_n+l)}, .., s_{(q_n \times (q_m-1)+l)}$.

Fig. 5(b) shows an example on the above designs. Assume that (i) D_x does not decide a parent device and D_y has decided a parent device and (ii) $C_L = 40\%$ and $C_S = 20\%$. According to the first assumption, D_x and D_y adopt Rule 1 and Rule 2 to decide their slots, respectively. Then, by the second assumption, q_m and q_n are 3 and 5, respectively. In this example, device D_x first decides $k = 1$, and D_x sets slots numbered 5 .. 9 for its listen state (as specified in Rule 1). By the assignment, we can see that D_x selects a row of slots for its listen state (as shown in the left part of Fig. 5(a)). On the other hand, device D_y decides $l = 3$, and D_y sets slot numbered 3, 8, and 13 for its search state (as specified in Rule 2). We can see that D_y selects a column of slots for its search state (as shown in the right part of Fig. 5(a)).

Theorem 2: After applying Rule 1 and Rule 2, the duty cycle requirement C_L and C_S can be satisfied.

Proof: In the above Rule 1, a device chooses slots in one row to perform its listen state operation. After applying Rule 1, the duty cycle of the device will be $\frac{q_n}{q_m \times q_n} = \frac{1}{q_m}$. From Eq. (2), we can know that $q_m \geq \frac{1}{C_L}$, and thus $\frac{1}{q_m} \leq C_L$. The selection by Rule 1 can satisfy the duty cycle requirement C_L. Moreover, by Rule 2, a device chooses slots in one column to perform its search state operation. By Eq. (3), $q_n \geq \frac{1}{C_S}$, and thus $\frac{1}{q_n} \leq C_S$. □

Recall that in our system, D_R continuously broadcasts its probe request packets and processes probe response packets from devices without entering sleep mode. In the following, we describe the detailed operations of a device $D_v \in \{\overline{V} \backslash D_R\}$. When emergency, initially, D_v does not have a parent device. At this time, D_v decides its slots for its listen state by the above Rule 1. After deciding slots, the schedule of D_v will be (i) listening to probe request packets in those slots decided by Rule 1 and (ii) entering sleep mode on the remaining slots. Then, D_v can start its first slot, and repeats its schedule every time frame. Moreover, assume that D_v receives a probe request packet in its listen state. At this time, D_v records the corresponding slot and takes the sender as its parent. Then, D_v can send message to its parent device through probe response packets. (Note that if D_v receives multiple probe request packets in its listen state, D_v can select a probe request sender, whose quality is better, to be its parent device.)

After deciding a parent device, device D_v can also become a parent device by applying the above Rule 2 to decide slots for its search state. After applying Rule 2, D_v changes its schedule to be (i) listening to its parent's probe request packets on the specified slot (that received its parent's first probe request packet), (ii) broadcasting its probe request packets continuously in those slots decided by Rule 2, and (iii) entering sleep mode on the remaining slots. During the search state, if D_v receives a device's probe response packet to it, D_v will take the sender as its child device. Device D_v also helps to relay messages from its child devices to its parent device. Furthermore, if D_v realizes that it has no child devices after performing search state operation for a period of time, D_v can give up to perform search state to save its power. In other words, D_v will be a leaf device in the network, and it only needs to wake up to listen to its parent device's probe request packets.

Theorem 3: If the network is connected, the proposed distributed scheme can guarantee there is no orphan device in the network.

Proof: Based on our network model in Section IV and the proposed distributed scheme, devices are logically

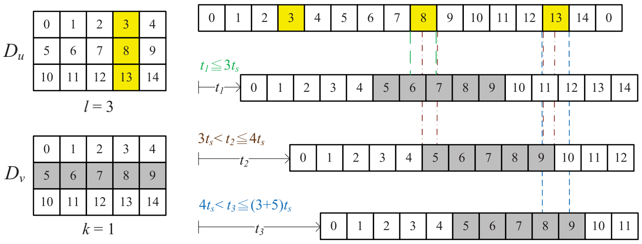

FIGURE 6. An example of the proof of Theorem 3.

connected by a tree topology rooted at D_R. Each device D_v can have at least one neighbor device D_u, which is located closer to the root, and the neighbor device D_u can become D_v's parent device. Assume that the D_u starts its search state in l-th column in D_u's grid quorum (or say D_u's time frame), and D_v executes its listen state operation in k-th row in D_v's grid quorum. In the following, we show that D_v's listen state can always overlap with D_u's search state. First, if D_u and D_v are time synchronized, D_v and D_u can meet at their common slot numbered $s_{k\times q_n+l}$. Second, if the start timing of D_v's time frame is t time unit later than D_u's, we can have the following four cases. (i) If $t \leq l \times t_s$, in this case, D_v can meet D_u during D_v's slots numbered $s_{k\times q_n+l-t_f-1}$ and $s_{k\times q_n+l-t_f}$, where $t_f = \lfloor \frac{t}{t_s} \rfloor$. (ii) If $l \times t_s \leq t \leq (l+1) \times t_s$, in this case, D_v can meet D_u twice at D_v's first and last listen state slots, i.e., D_v's slot numbered $s_{(q_n\times k)}$ and $s_{(q_n\times(k+1)-1)}$. (iii)If $(l+1)\times t_s < t \leq (l+q_n)\times t_s$, in this case, D_v can meet D_u at D_v's slots numbered $s_{(q_n\times(k+1)-1+l-t_f)}$ and $s_{(q_n\times(k+1)+l-t_f)}$, where $t_f = \lfloor \frac{t}{t_s} \rfloor$. (iv) If $t > (l+q_n)\times t_s$, this case means that the time drift has been exceeded the column size of the grid quorum. By setting $t = t\%t_s$, this case will be the same as cases (i)(ii)(iii) above. ☐

Fig. 6 indicates an example of the above cases. In the example, the parameters $l = 3$ and $k = 1$. The drift t_1, t_2, and t_3 are about 1.5, 3.5, and 4.5 slots. We can see that by our scheduling method, D_v's listen states can meet D_u's search states no matter the drift of D_v's clock.

Theorem 4: The network discovery latency $L_D = O(2 \times M_{dep} \times T_F)$, where $T_F = t_s \times q_m \times q_n$ and M_{dep} is the network depth of G.

Proof: By the distributed scheme, after D_R starts broadcasting its probe request packets, it is not hard to see that the devices that located in M_{dep} can be discovered after M_{dep} time frames, i.e. T_F time units. Then, the message from devices in M_{dep} also need M_{dep} time frames to relay to D_R. ☐

1) ENHANCEMENT ON REDUCING LATENCY

According to the above design, latencies on downstream and upstream data are both $O(M_{dep} \times T_F)$ units of times. In the following, we make two modifications to further reduce latencies. The first modification is to carry time information of the probe request packet sender. More specifically, in a probe request packet, the sender, say D_v, carries its current slot number s_i and a t_{diff} value, which records the time difference between the current time and the start of D_v's current time frame. By the carried s_i and t_{diff} in D_v's probe request packet, a device D_u that receives D_v's probe request packet can align its time frame to the one of D_v. After the alignment, the time frames of between D_v and D_u can be synchronized loosely. (Note that for the D_R, it always sets s_i to be s_0.)

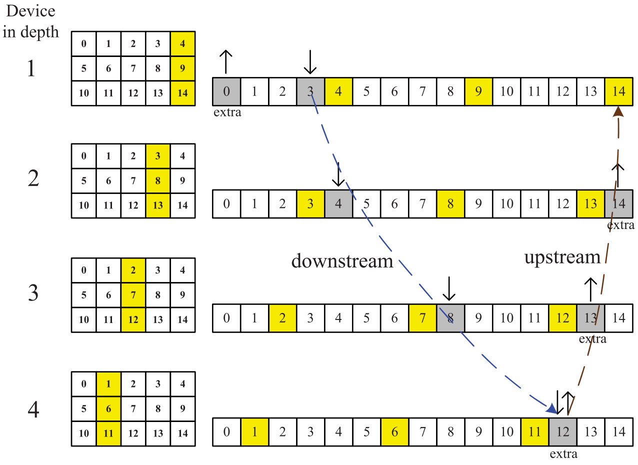

FIGURE 7. An example of scheduling on the enhancement scheduling.

The second modification is to revise the original Rule 2 of the distributed scheme. Again, by the slot information s_i in D_v's probe request packet, a device D_u can infer the D_v takes l-th column of the grid quorum for D_v's search states. By l, we calculate a value $l' = (l-1)\%q_n$. For a device D_u that takes D_v as its parent device, the modified Rule 2 is as follows.

- Rule 2′: The device D_u takes slots in l'-th column of the grid quorum as its search state. Then, D_u also wakes up to execute listen state operations in slots numbered s_i and an extra slot numbered $s_{(q_n\times(q_m-1)+l)\%(q_n\times q_m)}$.

Fig. 7 shows an example of the above enhancement. Assume that there are four devices that are located in depth 1 to depth 4 in he network. The device in depth 1 takes D_R as its parent, the device in depth 2 takes the device in depth 1 as its parent, and so on. By the first modification, the time frames of devices are aligned (loosely). Then, by the above Rule 2′, a device chooses a column of slots (which are prior to its parent's) for its search state. As shown in Fig. 7, by Rule 2′, each device further selects an extra slot for listen state, which can facilitate reporting upstream data quickly. Our modifications favor upstream data because that when emergency, affected people may be demanded to report periodically, and thus the modifications can facilitate tracking affected people's statuses. In this design, devices can still rely on the first listen state to disseminate downstream data with bounded latency. Let L_b represent the maximum downstream latency from D_R to any device and L_r represent the maximum report latency on user messages.

Theorem 5: By the above enhancement, $L_b = O(q_n \times t_s \times M_{dep})$.

Proof: Assume that device D_v receives a downstream data packet from its parent device at slot numbered s_i.

According to the assignment rule, D_v can send the data packet to its child device on its next search state slot, i.e., the next $q_n - 1$ slot of s_i. So, the one hop downstream latency will be $O(q_n \times t_s)$. As a result, the L_b can be dominated by $O(q_n \times t_s \times M_{dep})$. □

Theorem 6: By the above enhancement, $L_r = O(t_s \times M_{dep} + T_F)$, where $T_F = t_s \times q_m \times q_n$.

Proof: For the best case, when a device D_v receives a message from its child device at a slot s_i, the message can be relayed to D_v's parent at the next slot of s_i. Thus, the best L_r can be $O(t_s \times M_{dep})$. However, for any report that generated at any time, the report data can be relayed by the best case schedule within at most $q_m \times q_n$ slots, i.e., a time frame T_F. As a result, the L_r will be $O(t_s \times M_{dep} + T_F)$. □

VI. SIMULATION RESULTS

In this work, we develop a simulator (using Python) to verify the proposed schemes. In our simulations, there are N devices randomly distributed in an $R \times R$ meters square region, and one device is selected as the D_R of the network. Devices are assumed to have the same transmission range T_r meters. In the following, the proposed centralized algorithm, distributed quorum-based algorithm, and enhanced QO algorithm are denoted as CN, QO, and QE, respectively. The slot size of CN, i.e., t_o and the slot size of QO/QE, i.e., t_s, are the same. In QO and QE, the duty cycle requirements C_L and C_S are directly translated to the settings of q_m and q_n. For comparison purpose, in CN, the size of a time frame will the same as the quorum size of QO and QE, i.e., $(q_m \times q_n)$.

We compare the proposed CN, QO, and QE schemes against a distributed random-based algorithm (denoted by RN). In the RN, the network time is also divided into slots, and $(q_m \times q_n)$ slots form a time frame. When a device does not find a parent, this device will stay in listen state to wait probe request packets. After deciding a parent device, this device will then randomly choose q_m slots as its search states, and keep tracking its parent's search states for one slot.

In our simulation, we measure the network discovery latency L_D (defined in Definition 2) under different network settings. Besides, based on the decided time slots for listen and search states (using CN, QO, QE, and RN), we further measure (i) the maximum downstream latency L_b from D_R to any device and (ii) the maximum report latency L_r on user messages. The values of L_D, L_b, and L_r are recorded in number of slots. Moreover, each data point in the simulation results is the average of 100 trials conducted on 100 different randomly generated networks. The used notation in our simulations are summarized in Table 1.

TABLE 1. The used notations in our simulations.

Parameter	Notation	Unit
Number of devices	N	
Network size	$R \times R$	m^2
Transmission range	T_r	meters
Network density ratio	K	
Time frame size	$q_m \times q_n$	slots
Network discovery latency	L_D	slots
Maximum downstream latency	L_b	slots
Maximum report latency	L_r	slots

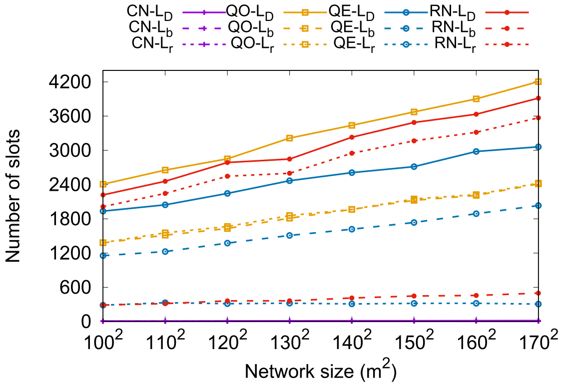

FIGURE 8. Simulation results on the effects of network size.

First, we observe the effect of different network size by varying parameter R. In this simulation, we fix $T_r = 25$ meters, $q_m = 10$, $q_n = 50$, and there will have $N = R^2/100$ devices. Fig. 8 shows the simulation result. From the result, we can see that when the network size becomes larger, L_D, L_b, and L_r values increase accordingly. The CN can outperform QO, QE, and RN in L_D, L_b, and L_r since CN is a centralized scheme. From the result, RN will have a smaller L_D than QO. This is because that devices using RN stay in listen state when they do not decide a parent, and thus devices using RN can find parents more quickly. Again, the QE also has a shorter L_D than QO and RN. This is because that a device using QE can be discovered within q_n slots with the help of extra slots for listen state. Moreover, we can see that L_b and L_r values of QO are similar because that devices have the same chances on bumping into parent devices' search states and child devices' listen states. The RN induces a smaller L_b than QO and QE. This is because that in RN, the reserved slot for listen state favor downstream data only. On the other hand, QE favors the upstream data, and thus the QE can have a smaller L_b than QO and RN.

Next, we observe the effects when changing devices' transmission ranges. In this simulation, we fix $R = 200$ meters, $N = 400$, $q_m = 10$, and $q_n = 50$. Fig. 9 indicates the result. We can see that when transmission ranges become larger, the latencies will be smaller because of the reduced maximum network depths. Again, the CN can outperform the other schemes. The other three schemes follow the similar trends as the ones in Fig. 8. Moreover, in Fig. 10, we observe the effects on varied network density values. In this simulation, we fix $R = 200$ meters, $Tr = 25$, $q_m = 10$, and $q_n = 50$. We vary the $N = K \times 100$, where K is the network density ratio, i.e., if the K becomes larger, the density of the network will be higher. From the result, the changes on network density do not affect L_D, L_r, and L_b significantly. This is because that since the transmission ranges of devices are not increased, the network depths will not be shortened when the density becomes higher.

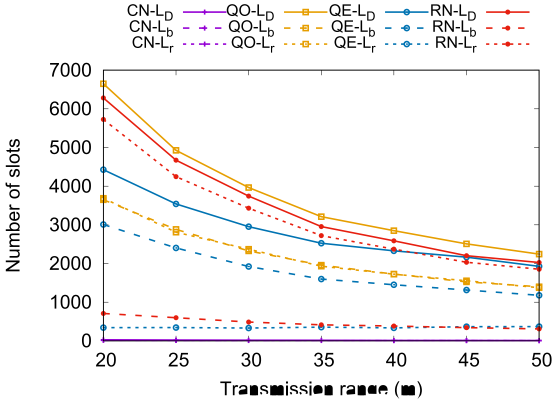

FIGURE 9. Simulation results on the effects of transmission range.

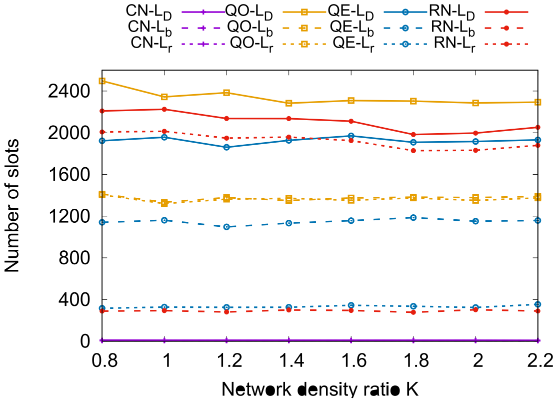

FIGURE 10. Simulation results on the effects of network density.

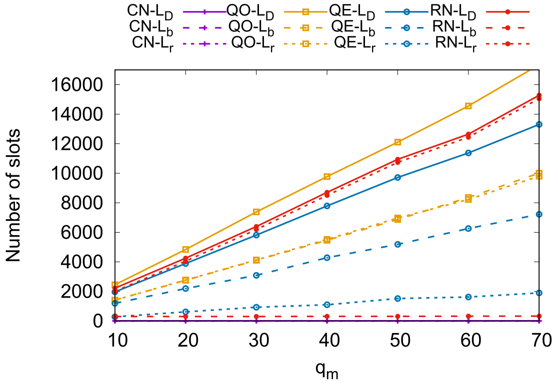

FIGURE 11. Simulation results on the effects of parameter q_m.

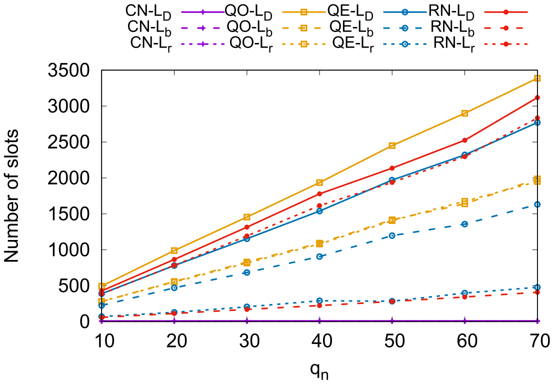

FIGURE 12. Simulation results on the effects of parameter q_n.

Next, Fig. 11 and Fig. 12 indicate the effect on changing the size of time frames by varying parameters q_m and q_n. In these two simulations, we fix $R = 100$ meters, $N = 100$, and $Tr = 25$. The default q_m and q_n are set to 10 and 50, respectively. We can see that CN does not affect by the changes of time frame sizes. For QO, QE, and RN, most L_D, L_b, and L_r will increase accordingly when q_m and q_n become larger. The L_b of RN can perform better when q_m becomes larger because that devices in RN can select more slots for search state operation. We can also see that the L_r of QE only increases slightly when q_m and q_n become larger. This result indicates that the design of the QE can effectively arrange devices search state and listen state to facilitate data reporting.

In summary, from the above simulations, the CN can perform the best, but CN is a centralized scheme. The QO, QE, and RN are distributed schemes, which are more suitable for real cases. The L_D of RN can be slightly better than QO because that each device in RN stays in listen state before deciding a parent device, but in QO and QE, devices follow the duty cycle requirements to save battery power. In those three distributed schemes, QE can perform the best in most cases by carrying time information in probe request packet and using one more slot for listen state after devices being discovered.

VII. IMPLEMENTATION RESULTS

A. IMPLEMENTATION USING ANDROID Wi-Fi DIRECT API

In this work, we modify the original Android Wi-Fi Direct API source code [6] to implement the designed QO scheme. The used Android version is 6.0.1 from the Android Open Source Project website [2], and the modified source code is executed on Google Nexus 7. Fig. 13 shows the implemented and modified modules in the Android source code. Our implementations are located in three different Android layers, i.e., the APP layer, the Android framework layer, and the driver control layer.

1) THE DRIVER CONTROL LAYER

In the driver control layer, the ctrl_iface module is responsible for receiving commands from WifiP2pServiceImpl class. After receiving a command, the ctrl_iface class first checks if command format is correct. If so, it will call p2p_supplicant class to process the command. Then, the p2p_supplicant class extracts the command, and transfers the command to the p2p module. In our implement, we modify the ctrl_iface and p2p_supplicant classes to process our commands to support QO.

In this work, we mainly did five modifications on the original *p2p module*. First, we modify the device discovery procedure to skip the scan phase and to execute the find phase

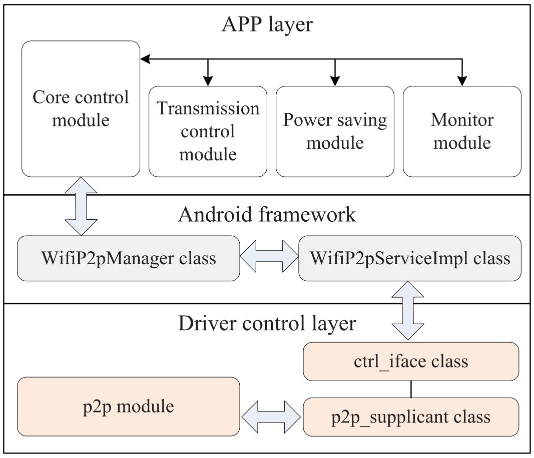

FIGURE 13. The implemented and modified modules in Android.

directly. Second, during the find phase, the p2p module will be controlled by commands from the APP layer to switch between listen state and search state. (The command is sent according to the schedule decide by the QO scheme.) Third, in the original p2p module, when a device receives a probe request packet or a probe response response packet, this module will be informed by an internal *device_found* notification. But, for the same probe request/response packet sender, the notification will be triggered only once. So, we modify to trigger device_found notifications multiple times since our design relies on probe request and response packets to communicate. Fourth, when obtaining a device_found notification, this module extracts the device name field in the received probe request/response packets. Then, this module sends the extracted data to the upper layer through p2p_supplicant and ctrl_iface classes in the driver control layer. Fifth, to compatible with the original Android implementation, we further implement a launcher in this layer. So, this layer can be *normal mode*, which executes the original Wi-Fi direct functionalities, or can be *emergency mode*, which operates the implemented QO scheme.

2) THE ANDROID FRAMEWORK LAYER

In the Android framework layer, the WifiP2pManager and WifiP2pServiceImpl classes are responsible for receiving commands from the APP layer and relaying commands to the driver control layer, respectively. We add commands to facilitate interactions between the APP layer and driver control layer when emergency. The added commands include (i) the controls of starting listen and search states, (ii) carrying emergency messages, and (iii) the switch between normal mode and emergency mode on the driver control layer. We also modify the corresponding interface APIs to support the delivery on the implemented commands.

3) THE APP LAYER

In the APP layer, all modules are newly implemented. First, the core controller module controls flows of the designed system. Besides, this module maintains the device's parent and the list of child devices. This module utilizes existing Android APIs to control other modules and to send commands to the lower layer. Second, the power saving module controls the switch between search state, listen state, and sleep state. Third, the data transmission module implements the functionality of transmitting information between device through probe request and response packets. This module also handles the processing of the designed ACK/NACK signals and performs retransmissions when necessary. Forth, the monitor module is implemented based on the Android broadcast receiver service.

4) THE DETAILS ON THE REVISED PROBE REQUEST/RESPONSE PACKETS

In our implementation, devices utilize the exchange of probe request and probe response packets for emergency communications. We utilize the "device name" field (with size 32 bytes) in the probe request and probe response packets to carry the information transmitted between devices. When filling data in a predefined space, we bump into two implementation issues. The first issue is that the expression on a "character" of Java and C programming languages are different. According to the original Android design, the programming languages used in the APP layer and the Android framework layer are Java and C, respectively. The Java programming language uses 2 bytes to express a single character, but the C programming language uses 1 byte. When expressing a character with 7-th bit set to 1, the Java programming language will automatically set all digits in its higher byte to be 1. For example, given a character 0xAC, the Java programming language translates the character to be 0xFFAC. But, when C programming language receives the 0xFFAC, it will consider that there are two characters, i.e., 0xFF and 0xAC, respectively. Second, based on our experiments, there will have random errors between the APP layer and Android framework layer if the value of a Java character is less than 32. The reason on the random errors is still unknown to us. In this work, to resolve the above two issue, in the APP layer, we force the 7-th bit and 6-th bit of the lower byte in Java to be 0 and 1, respectively. So, by the restriction, for a byte space, we can use only 6 bits to store data.

Fig. 14(a) shows the format of the modified device name field in the probe response packet. The total data payload in a probe response packet can be 16 bytes, and 96 bits (i.e., 16 × 6 bits) are available to carry an affected individual's message. A probe response packet sender can divide a message (from an affected individual) into multiple probe response packets. In our design, a device can use at most six probe response packets to carry a message. Those six probe response packets will be labeled by sequence numbers from 0 to 5. So, in the total/now field, we use 3 bits for total number of used probe responses of a message and 3 bits for the sequence number. Moreover, Fig. 14(b) shows the format of the modified device name field in the probe request packet. A probe request packet can be a data or control packet

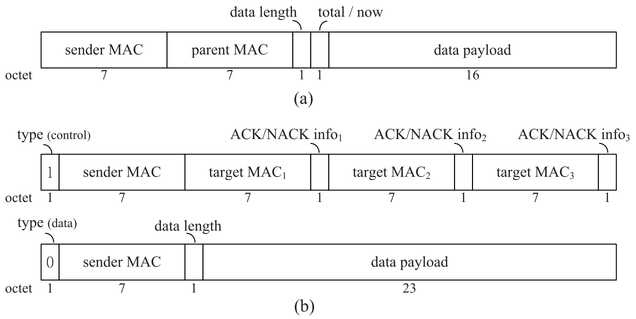

FIGURE 14. **The modified device name field in (a) probe response packet and (b) probe request packet.**

(depend on the first bit of the device name field). The D_R or a parent device checks if the received packet is in sequence, and then carry corresponding devices' MAC address and ACK/NACK information in the next probe request packet. A parent device can carry ACK and NACK information for at most three child devices. Furthermore, in our design, we use only 42 bits to express devices' MAC addresses. According to the rule of the MAC address, the length of a MAC address should be 6 bytes (i.e., 48 bits), where 3 bytes are as manufacturer labels and 3 bytes are the network address specified by the manufacturer. However, we found that 18 bits are enough to express all the manufacturers. In our implementation, we use a table in the APP layer to map real manufacturer labels and the modified manufacturer labels. As a result, we can use only 42 bits to express a MAC address.

B. EXPERIMENT RESULTS

In our experiment, we further implement a reduced version of QO (denoted by QO-R) for comparisons. The QO-R simply utilizes the original Wi-Fi direct device discovery procedure as its lower layer. More specifically, QO-R only applies our APP layer functions, and the Android source codes are unchanged. In QO-R, the scan phase in the original Wi-Fi device discovery procedure cannot be omitted. After entering the find phase, the lower layer has to alternatively execute search state and listen state. We configure the lower layer to execute search state and listen state for 2 seconds and 2 seconds repeatedly. Moreover, as we mentioned above, in the original Android implementation, when a device receives multiple probe request or probe response packets from a neighbor device, only the first packet from the neighbor will be sent to the device's APP layer. So, in order to allow multiple probe request/response packets from the lower layer, in QO-R, the APP layer will reset the device discovery procedure when receiving a probe request/response packet.

In the following experiments, we set quorum size parameters $q_m = 10$ and $q_n = 6$, and set the slot size parameter $t_s = 1$ minute. We first compare the one hop transmission latency of QO and QO-R. In this experiment, there are two devices, and a child device generating 1 to 9 packets to its parent device. The transmission latency is measured as the time between the first packet sent by the child device's APP layer and the last packet received by the parent device's APP layer. Fig. 15 shows the results. We can see that when the child device needs to send more packets, the transmission latencies increase accordingly. The QO can outperform QO-R in most cases. This is because that the parent device of QO-R has to reset the device discovery procedure every time when receiving one probe request packet from the child device. Based on our experiments, it takes about 30 to 70 ms to reset the device discovery procedure. As a result, the transmission latency of QO-R will be higher.

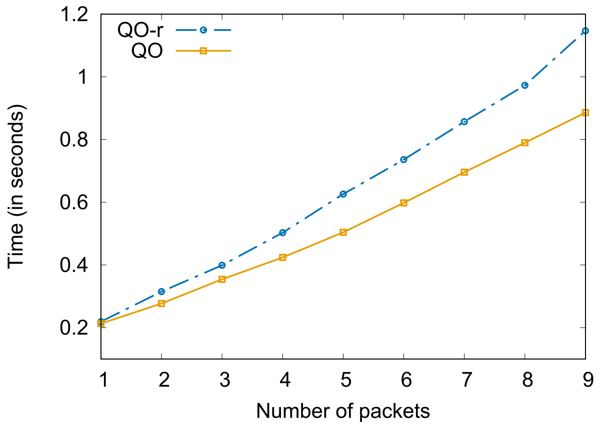

FIGURE 15. **Comparisons of the one hop transmission latency of QO and QO-R.**

Next, we compare the energy consumption of QO and QO-R. In this experiment, there are three devices (D_R and two devices, say D_v and D_u). The device D_v takes D_R as its parent device, and we say that D_v is an in-tree device. Then, D_u takes D_v as its parent device, and we say that D_u is the leaf device. Recall that in our design, if a device does not have a child device, it will give up to perform the search state operations. So, the leaf device only wakes up for one slot in every time frame. Moreover, we further compare the battery capacity when a device only stays in standby mode (label as standby), keeps operating listen state (label as only_listen), and keeps operating search state (label as only_search). Before performing the experiment, devices' battery capacities are charged to 100%. In the experiment, a device sends four probe response packets to its parent device in each time frame, and we record the trend of the remaining battery capacity for 24 hours.

Fig. 16 shows the result. We can see that the battery capacity will be exhausted within 16 hours if the device only executes the search state operation. Compare to only_search, the energy of the in-tree device can be greatly preserved. Again, compare to only_listen, the proposed scheme can also reduce energy consumption of the leaf device. The device stays in standby mode can have the least energy consumption but the device cannot be found when emergency. Moreover, by the result, we can see that the QO and QO-R have similar performance on energy consumption because that QO and QO-R apply the same wake up and sleep schedule. But, the QO is still slightly better than QO-R since the devices in QO-R need to repeatedly switch listen and search states when waking up, which will induce some energy consumption.

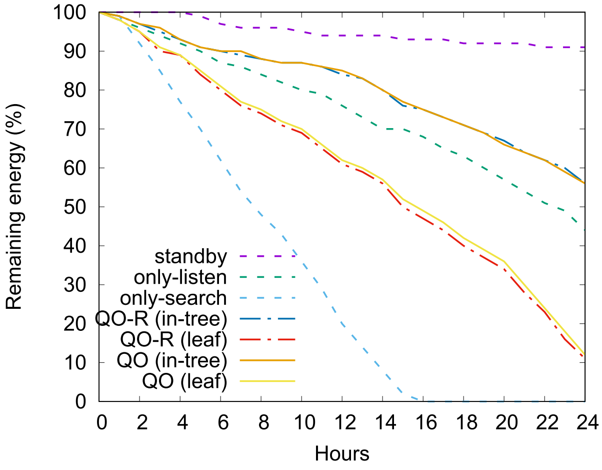

FIGURE 16. Comparisons of the energy consumption of QO and QO-R.

C. DEMONSTRATION

We show our implementation result by a video clip. We implement a rescue APP and a status reporter APP for rescuer and for affected people. The rescue APP can start service by broadcasting probe request packets. After receiving probe request packet, the status reporter APP will report the owner's basic information to the rescue device. After the rescue device replies ACK signal, the status reporter APP can then report the current status (e.g., bleeding, feel hurt, and so on) of the owner to the rescue device. For the demonstration, please refer to [3].

VIII. CONCLUSIONS

In this paper, we design group-less and energy efficient communication schemes based on the Wi-Fi direct technology for emergency scenes. In the network, devices are connected by a tree topology, and use the modified probe request and response packet to carry messages from affected people or rescue personnel. We observe that when emergency, devices have to execute sleep scheduling to save their battery power and all devices should be discovered. So, we model the target network scenario by an EWDS problem and then propose a centralized solution and a distributed quorum-based solutions for devices to schedule their wake up and sleep schedule. To facilitate data reporting, we further design an enhancement on the distributed scheme. The simulation results indicate that compare to a random scheduling method, the proposed schemes can effectively shorten network discovery latency and devices' report latency. The prototyping results further indicate that the proposed distributed scheme can effectively operate on commercial smartphones and can save smartphones' battery power. In the future, we are going to use drones to provide dynamic emergency network formation service.

REFERENCES

[1] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Standard 802.11b-1999, 2012.

[2] *Android Open Source Project*. Accessed: Mar. 03, 2019. [Online]. Available: https://source.android.com/

[3] *The Demo Video of Our Implementation*. Accessed: Mar. 03, 2019. [Online]. Available: https://www.youtube.com/channel/UCbg3RaqCueuYPjaZFc85oCA

[4] *2010 Haiti Earthquake*. Accessed: Mar. 03, 2019. [Online]. Available: https://en.wikipedia.org/wiki/2010_Haiti_earthquake

[5] *Wi-Fi Peer-to-Peer (P2P) Technical Specification, Wi-Fi Alliance Technical Committee P2P Task Group*, document v.1.5, 2014.

[6] *Android Wi-Fi Peer-to-Peer API*. Accessed: Mar. 03, 2019. [Online]. Available: https://developer.android.com/guide/topics/connectivity/wifip2p

[7] G. Aloi *et al.*, "Stem-net: How to deploy a self-organizing network of mobile end-user devices for emergency communication," *Comput. Commun.*, vol. 60, pp. 12–27, Apr. 2015.

[8] G. Araniti, A. Raschellà, A. Orsino, L. Militano, and M. Condoluci, "Device-to-device communications over 5G systems: Standardization, challenges and open issues," in *5G Mobile Communications*. Cham, Switzerland: Springer, 2017, pp. 337–360.

[9] L. Baresi, N. Derakhshan, S. Guinea, and F. Arenella, "MAGNET: A middleware for the proximal interaction of devices based on Wi-Fi direct," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–7.

[10] C. E. Casetti, C. F. Chiasserini, Y. Duan, P. Giaccone, and A. P. Manriquez, "Data connectivity and smart group formation in Wi-Fi direct multi-group networks," *IEEE Trans. Netw. Service Manag.*, vol. 15, no. 1, pp. 245–259, Mar. 2018.

[11] C.-M. Chao and Y.-W. Lee, "A quorum-based energy-saving MAC protocol design for wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 59, no. 2, pp. 813–822, Feb. 2010.

[12] G. Chatzimilioudis, C. Costa, D. Zeinalipour-Yazti, and W.-C. Lee, "Crowdsourcing emergency data in non-operational cellular networks," *Inf. Syst.*, vol. 64, pp. 292–302, Mar. 2017.

[13] F. R. Dogar, P. Steenkiste, and K. Papagiannaki, "Catnap: Exploiting high bandwidth wireless interfaces to save energy for mobile devices," in *Proc. 8th Int. Conf. Mobile Syst., Appl., Services (MobiSys)*, 2010, pp. 107–122.

[14] C. Funai, C. Tapparello, and W. Heinzelman, "Enabling multi-hop ad hoc networks through WiFi direct multi-group networking," in *Proc. IEEE Int. Conf. Comput., Netw. Commun. (ICNC)*, Jan. 2017, pp. 491–497.

[15] T. Furutani, Y. Kawamoto, H. Nishiyama, and N. Kato, "A novel information diffusing method with virtual cells based Wi-Fi direct in disaster area networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2018, pp. 1–6.

[16] L. Hamidouche, P. Sens, S. Monnet, and D. Refauvelet, "Toward heterogeneity-aware device-to-device data dissemination over Wi-Fi networks," in *Proc. IEEE Int. Conf. Parallel Distrib. Syst. (ICPADS)*, Dec. 2017, pp. 105–112.

[17] O. D. Incel, A. Ghosh, B. Krishnamachari, and K. Chintalapudi, "Fast data collection in tree-based wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 11, no. 1, pp. 86–99, Jan. 2012.

[18] K. Jahed, O. Farhat, G. Al-Jurdi, and S. Sharafeddine, "Optimized group owner selection in WiFi direct networks," in *Proc. IEEE Int. Conf. Softw., Telecommun. Comput. Netw. (SoftCOM)*, Sep. 2016, pp. 1–5.

[19] Y. Jahir, M. Atiquzzaman, H. Refai, A. Paranjothi, and P. G. LoPresti, "Routing protocols and architecture for disaster area network: A survey," *Ad Hoc Netw.*, vol. 82, pp. 1–14, Jan. 2019.

[20] W.-S. Jung, H. Ahn, and Y.-B. Ko, "Designing content-centric multi-hop networking over Wi-Fi direct on smartphones," in *Proc. IEEE Int. Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2014, pp. 2934–2939.

[21] G. Z. Khan, E.-C. Park, and R. Gonzalez, "M^3-cast: A novel multicast scheme in multi-channel and multi-rate WiFi direct networks for public safety," *IEEE Access*, vol. 5, pp. 17852–17868, 2017.

[22] M. A. Khan, W. Cherif, F. Filali, and R. Hamila, "Wi-Fi direct research—Current status and future perspectives," *J. Netw. Comput. Appl.*, vol. 93, pp. 245–258, Sep. 2017.

[23] P. Kolios, C. Panayiotou, and G. Ellinas, "Disaster resilient telematics based on device-to-device communication," *Wireless Public Saf. Netw.*, vol. 3, pp. 211–230, Jan. 2017.

[24] E. Kuada and B. Bannerman, "Opportunistic rescue network for disaster management," in *Proc. IEEE AFRICON*, Sep. 2017, pp. 917–922.

[25] E. K. Lee, H. Viswanathan, and D. Pompili, "RescueNet: Reinforcement-learning-based communication framework for emergency networking," *Comput. Netw.*, vol. 98, pp. 14–28, Apr. 2016.

[26] T. Lee, K. Han, M. Lee, S. Park, and C.-Y. H. Yang, "Multi-connectivity strategy for agile service delivery with low power discovery," in *Proc. IEEE 25th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Sep. 2014, pp. 2071–2075.

[27] Q. Li and D. Rus, "Global clock synchronization in sensor networks," *IEEE Trans. Comput.*, vol. 55, no. 2, pp. 214–226, Feb. 2006.

[28] M.-S. Pan and Y.-H. Lee, "Fast convergecast for low-duty-cycled multi-channel wireless sensor networks," *Ad Hoc Netw.*, vol. 40, pp. 1–14, Apr. 2016.

[29] A. A. Shahin and M. Younis, "Alert dissemination protocol using service discovery in Wi-Fi direct," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2015, pp. 7018–7023.

[30] F. Sivrikaya and B. Yener, "Time synchronization in sensor networks: A survey," *IEEE Netw.*, vol. 18, no. 4, pp. 45–50, Jul. 2004.

[31] M. Usman, M. R. Asghar, I. S. Ansari, F. Granelli, and K. Qaraqe, "Towards energy efficient multi-hop d2d networks using wifi direct," in *Proc. IEEE Int. Global Commun. Conf. (Globecom)*, Dec. 2017, pp. 1–7.

[32] W. Wang, Y. Chen, and L. Wang, "Sampleless Wi-Fi: Bringing low power to Wi-Fi communications," *IEEE/ACM Trans. Netw.*, vol. 25, no. 3, pp. 1663–1672, Jun. 2017.

[33] S. Xiao and W. Li, "Mixed power saving mode for D2D communications with Wi-Fi direct," in *Proc. Int. Conf. Mechatron. Intell. Robot. (ICMIR)*, 2017, pp. 234–240.

[34] C. Yao, H. Zhang, and L. Song, "WiFi multihop: Implementing device-to-device local area networks by android smartphones," in *Proc. ACM Int. Symp. Mobile Ad Hoc Netw. Comput. (MobiHoc)*, 2015, pp. 405–406.

[35] M. Yun, D. Kim, H.-S. Lee, and J. Lee, "Silent broadcast: Experience of connectionless messaging using Wi-Fi P2P," in *Proc. IEEE 8th Int. Conf. Inf. Sci. Digit. Content Technol. (ICIDT)*, Jun. 2012, pp. 239–242.

MENG-SHIUAN PAN received the Ph.D. degree from National Chiao Tung University, Taiwan. He was with HTMM Corporation and dedicated in designing and implementing 3GPP layer 2/3 protocol stacks. He is currently an Associate Professor with the Department of Computer Science and Information Engineering, Tamkang University, Taiwan. His research interests include mobile computing, wireless communication, and the Internet of Things. He serves as an Associate Editor of the *International Journal of Distributed Sensor Networks*.

CHUNG-MING WANG received the B.S. and M.S. degrees from Tamkang University, Taipei, Taiwan. His research interests include wireless networks and protocol design.

• • •