Maintaining Connectivity in Ad Hoc Networks Through WiFi Direct

Utku Demir, Cristiano Tapparello, Wendi Heinzelman
Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY, USA
{udemir, ctappare, wheinzel}@ece.rochester.edu

***Abstract*—The wide diffusion of mobile devices that natively support ad hoc communication technologies has led to a number of protocols for enabling and optimizing Mobile Ad Hoc Networks (MANETs). Nevertheless, the actual utilization of MANETs in real life is still limited, in part due to the lack of protocols for the automatic creation and evolution of ad hoc networks. Recently, a novel ad hoc protocol named WiFi Direct has been proposed and standardized by the WiFi Alliance with the objective of facilitating the interconnection of nearby devices. WiFi Direct provides high performance direct communication among devices, includes different energy management mechanisms, and is now available in most modern mobile devices. However, the current WiFi Direct implementations require user interaction for setting up and maintaining the connection. In this paper, we propose and analyze three practical schemes for creating self-organizing and self-healing WiFi Direct networks using Android OS devices. Experimental results show that our proposed approaches are feasible with different overhead in terms of prior knowledge about the network and coordination between the devices. These techniques provide the first known approaches for the automatic creation and maintenance of MANETs using every day mobile devices.**

I. Introduction

Mobile Ad Hoc Networks (MANETs) are infrastructureless networks developed to meet the needs of a variety of applications where infrastructure-based wireless networks are difficult to deploy and maintain. MANETs represent a promising way to provide communication among the increasing number of mobile devices during disaster scenarios, military operations, or whenever devices are still expected to be able to communicate in an organized fashion in a challenging environment. Efficient ad-hoc communication standards, along with smart communication protocols able to scale to large numbers of devices, are therefore essential [1].

Several ad hoc communication protocols, such as IEEE 802.11 DCF, IEEE 802.11s, IEEE 802.11z, ZigBee, SMAC, Bluetooth, and WiFi Direct have been proposed in the literature. Even though these protocols are widely available for creating both single and multi-hop networks, in real life experiments, they generally show low performance or require high energy consumption. As an example, ZigBee and SMAC are designed for energy-constrained networks, but they only support low data rates, which might be inappropriate for large-scale MANETs. WiFi Direct (initially known as WiFi-P2P), on the other hand, has been identified as a prominent candidate for supporting communications in MANETs [2]. WiFi Direct aims to enhance WiFi based ad hoc communication, includes different energy management mechanisms, and is now available in most of the modern mobile devices [3], [4]. With WiFi Direct, devices are organized in groups, where one member of the group is the Group Owner (GO) and all the other devices are Group Members (GM). Being built on top of WiFi, groups are able to also support legacy clients, devices that do not support WiFi Direct but support WiFi. As a result, WiFi Direct enables ad hoc communication among WiFi devices, such as, for example, smartphones, tablets, laptops, and printers. However, it requires user interaction for setting up the ad hoc connection.

In addition to requiring an appropriate ad-hoc communication protocol, MANETs require smart protocols that are able to scale to large numbers of devices. To this end, clustering protocols have been widely recognized as a viable solution [5]. Many clustering protocols have been proposed in the literature, aiming at optimizing various metrics like, for example, energy efficiency and connectivity [5]. When considering mobility, as is the case for MANETs, most of the existing clustering protocols introduce a high overhead, since they require the nodes to continuously acquire and exchange connectivity information. In this regard, the complexity of different leader selection algorithms in distributed systems has been extensively studied [6]–[8]. Low-maintenance algorithms for cluster head selection, instead, are mostly based on a predefined metric or a time-varying probability [5], [9]. In addition, most of these clustering protocols have not been implemented and tested on real-life technology and their performance evaluations have been conducted through simulations, in part due to the limited diffusion of efficient ad hoc communication technologies.

Given the wide diffusion and promising performance of WiFi Direct, and its intrinsic cluster-based network organization, a number of studies have recently proposed WiFi Direct as a viable technology for enabling ad hoc communication among mobile devices [1], [10]–[14]. In particular, the authors in [1] provide an in-depth description of the protocol, and evaluate its performance under different settings through experimentation on real devices. Different practical methods for allowing the communication among devices that belong to different WiFi Direct groups has recently been proposed in [10], while a WiFi Direct-based multi-group data dissemination protocol for public safety has been proposed in [11]. A group formation algorithm based on WiFi Direct is presented in [14], where the authors simulate the performance of a random group formation protocol that selects the GO

according to a time varying probability.

In this paper, we explore different methods for allowing automatic WiFi Direct group creation and maintenance, and evaluate the performance using commercially available Android devices. In particular, we shed some light on the limitations of using the WiFi Direct standard for creating and maintaining mobile ad hoc networks, and we propose three schemes for the selection of the GO, which require different levels of prior knowledge about the network and coordination between the devices. For each GO selection and group (re)formation scheme, we discuss the tradeoffs in terms of time when implementing these different approaches on Android devices. Our approaches can be used for realizing self-organizing and self-healing WiFi Direct based MANETs.

The rest of the paper is organized as follows. Section II provides a brief overview of the WiFi Direct standard. Section III describes our proposed GO selection and group formation schemes. In Section IV we discuss our experimental results, evaluating the group formation time of the different methods considered. Section V concludes the paper.

II. WiFi Direct

WiFi Direct [15] is a standard released by the WiFi alliance that enables ad hoc communication between nearby devices, without requiring a wireless Access Point (AP). WiFi Direct utilizes IEEE 802.11 a/b/g/n infrastructure mode, and can transmit either at 2.4 GHz or 5 GHz.

During ad hoc communication, devices form a group where one of them is the Group Owner (GO) and all the others are considered Group Members (GM). It is important to note that these roles are not predefined but are negotiated during the construction of the group and remain fixed for the entire duration of the group. Additionally, WiFi Direct groups can also include standard IEEE 802.11 nodes that do not support WiFi Direct and are referred to as Legacy Clients (LC).

According to the standard [15], the GO represents an AP-like entity that provides basic service set (BSS) functionality and services for the associated clients. Acting as a soft AP, the GO advertises and allows nodes to join the group. The advertisement and group maintenance are performed through beacon packets, just like a typical IEEE 802.11 AP, and the GO is responsible for giving control of the channel to nodes in its network and routing data through clients in its group[1]. As a result, the group topology is a $1{:}n$ hierarchical structure, where multiple GMs and LCs are connected to one GO.

The nodes that support WiFi Direct go through a group formation process in order to determine the roles of the GO and the GMs. There are three group formation cases: standard, persistent and autonomous [15], [16]. During the standard group formation, the nodes discover each other to negotiate for the roles[2]. The persistent group formation process allows for a faster reconstruction of previous groups, since the GO negotiation phase is replaced by an invitation exchange, and the WPS Provisioning process is significantly reduced by reusing the stored network credentials. In autonomous group formation, a node assigns itself the role of GO and creates its own group. After the roles have been established, the devices go through a WiFi Protected Setup (WPS) Provision phase and, after completion, the GO assigns the GM and an IP address using the Dynamic Host Configuration Protocol (DHCP).

[1]Routing data between clients in a group is allowed but not defined by the standard.

[2]The nodes listen on channels 1, 6, and 11 in the 2.4 GHz band and, after finding another device, they negotiate as to which will act as the GO. This is done in a handshake process, where the devices exchange an *intent value*, and the device with the highest value becomes the GO.

III. GO Selection and Group Formation Schemes

In this section, we present our proposed group formation schemes, which are depicted in Figure 1. These algorithms are completely automatic, meaning that the devices are able to react and adapt to changes in network topology like, for example, node mobility or GO failure. Moreover, for each scheme the GO is responsible for sending connection invitations to the other available peers.

A. Backup-based Group Formation

The first scheme, called *Backup-based* Group Formation, uses a coordination between the existing GO and the GMs via socket messaging. During the first group formation the devices go to through the standard GO negotiation phase to determine the device that will act as the GO. After this connection, the GO acquires information about the other devices, and elects one them to act as the *backup GO* to allow the group to react to unpredictable failures

As soon as the current GO disconnects (i.e., the group is destroyed), the backup device declares itself as the new GO and, it starts discovering the other available peers and upon discovery invites them to join the group. All the other devices look for available peers and wait for a connection invitation from the backup GO. After the group is reformed, the algorithm is repeated from the beginning, with the new GO electing a new device to act as a backup. We note that in this group formation scheme, at any given time only one device can elect itself as the GO.

B. ID-Based Group Formation

The scheme starts with each device scanning nearby peers. Once the devices have discovered all the other devices around it each device compares its ID with the IDs of the other peers, and the device with the smallest ID (or, equivalently, the largest) is declared to be the new GO. The newly elected GO will then send invitations to the other devices, thus forming the group. Afterward, if the GO fails or moves, the other devices are in charge of reforming the group. We note that also in this group formation scheme, at any given time only one device can elect itself as the GO, since the GO selection is based on a pre-determined function of the device ID and every device ensures that all the surrounding peers have been found before starting the group formation process.

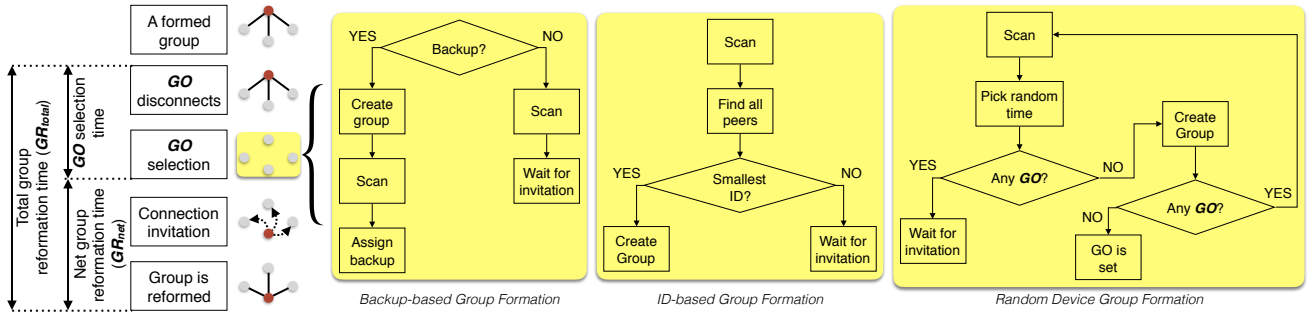


Figure 1: Performance metrics and flowcharts of the algorithms used for group maintenance.

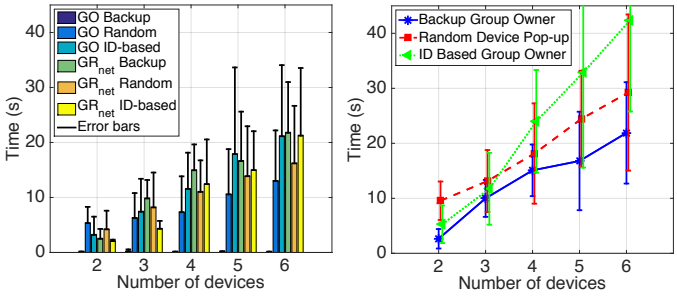
C. Random Device Group Formation

Unlike the other schemes, the *Random* device group formation includes a stochastic component for the selection of the GO. After the GO disconnects, each device starts a timer with duration obtained through a uniformly distributed random variable, and immediately starts discovering other peers. When the timer fires, the GO first looks for an existing GO and, if it doesn't detect any, it declares itself as the GO and starts sending connection invitations to all the other devices that it discovers in the vicinity. If when the timer fires the device finds an existing GO, instead, the device remains in a listening state and waits for a connection invitation. The process is repeated every time the devices detect changes in the network connection.

It is worth noting that multiple devices may elect themselves as GO, because both the group formation and the GO advertisement process requires a certain amount of time to be completed. Thus, during this time, a device might have started the group formation but it is not yet visible to the other peers. To limit the likelihood of this situation, the time interval used for the waiting period needs to be tuned according to both the WiFi Direct standard and the expected number of devices that will compete for the group formation. To this end, we define the vulnerable period T_v as the time interval between the start of the autonomous group formation process and the time at which the GO is visible to the other devices, and L to be the maximum waiting time (i.e., the waiting time is uniformly distributed in the interval $[0, L]$).

The probability of two GOs being simultaneously present is $Pr\{collision\} = 1 - (1 - T_v/L)^n$ where n is the number of devices that compete for forming the group. We note that the equation represents the probability that one device picks a waiting time less than T_v seconds apart from the smallest waiting time selected by the devices. Using the above equation we can determine the value of L that provides a desired probability of having multiple GOs.

When performing operations on real devices, a certain level of randomness in the vulnerable period T_v is introduced by the peer discovery process. Although this situation may be tolerated in large networks, in group formations involving a small number of devices, this should be avoided. Therefore, the *Random* scheme includes a step-back mechanism in order to ensure that, at any given time, only one GO is present in a given area of the network. According to the step-back mechanism, after a device declares itself as a GO, it checks if there is another GO visible as it is inviting the other peers. If a device detects another GO, it disconnects and re-starts the scheme, which eventually leads to having only a single GO in a given area of the network.



(a) GO change and net group reformation times.

(b) Total group reformation time.

Figure 2: Time required by the proposed group owner selection and group reformation schemes across number of devices.

IV. Performance Evaluation

In this section we evaluate the performance of the group formation schemes described in Section III, via implementation on identical Asus Nexus 7, running Android 4.4.2. For each group formation scheme, we measured the GO selection time and the total group reformation time, as shown in Figure 1, by considering groups with two to six devices. Moreover, to reduce the variability across experiments, we consider a persistent group formation with manual set-up of the initial group formation. Each experiment has been run 50 times.

A. GO Selection Time

GO change time depends on the number of devices in the network, as shown in Figure 2(a). The average GO selection time for the *Backup* scheme is around 0.15 seconds for each network size. Before the GO disconnects from the group, the backup GO has already been selected and can therefore elect itself as the GO as soon as it detects a change in connectivity.

In the *Random* scheme, upper bound L needs to be determined for its evaluation. To do so, we first performed repeated

experiments to determine the vulnerable period T_v, which is found to be an average value of $T_v = 1.5$ s. Setting the collision probability to 0.2 and inverting the equation in III-C, yields values of L equal to 14.208, 20.925, 27.6455, 34.366, and 41.087 s for two to six devices, respectively. As expected, the corresponding waiting interval increase is proportional to the number of devices in order to maintain a fixed probability of simultaneous GO selection. We note that the average GO selection time is more than seconds lower than the mean of the uniform distribution, because in order for a device to declare itself as a GO, it must pick the smallest number, which shifts the average GO selection time towards $L/(n+1)$ s.

Due to the stochastic nature of the discovery process, during our experiments simultaneous GO declarations occured. In particular, 25%, 18%, 20%, 16%, and 14% of the times two or more devices elected themselves as GO for the case with two to six devices, respectively. In the case of GO collisions, the step-back mechanism was able to ensure that only one GO was present in the network.

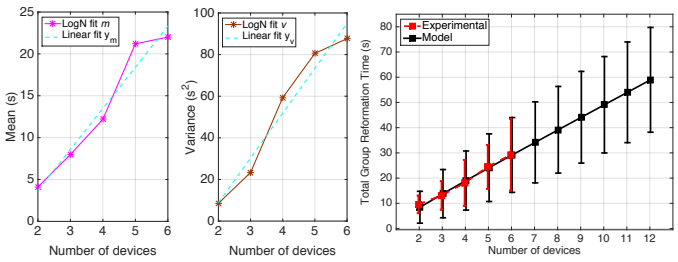
In addition, we ran 20 additional experiments to check the benefits of the step-back mechanism when setting $L = 7.5$ s for three devices. Even though we observed a higher rate of simultaneous GO selections, the resulting average GO selection time was reduced to 1.45 s, thus proving the efficacy of the step-back mechanism.

As shown in Figure 2(a), the performance of the *ID-Based* scheme is affected by the randomness of the discovery phase. This is because according to the *ID-Based* scheme, even if the GO selection is deterministic and it doesn't require any waiting time, the devices first need to discover all the other nearby peers. As a result, increasing the number of devices in the group requires more time to discover them, causing the average GO selection time to increase proportionally to the number of devices. Furthermore, the standard deviation of GO selection time increases as the number of devices increases due to the variability of the discovery phase. We note that in our experiments we assume that the total number of available devices is known by all the nodes, thus obtaining a lower bound on the actual performance of this scheme.

Given the above, the *Backup* scheme always has the smallest GO selection time, whereas the *Random* scheme tends to perform better than the *ID-Based* approach as the number of devices increases. Even though both the *Backup* and the *ID-Based* schemes make sure that there will not be any collisions in the selection of the GO, electing a backup device beforehand may not be useful in highly dynamic networks, while having to discover all the peers in the network is not practical for large networks. Thus, the *Random* GO selection with the step-back mechanism represents the more efficient solution for general WiFi Direct networks. This is also promising for large scale networks, since random algorithms have been shown to drastically reduce the leader election complexity [17].

B. Group Formation Time

We next evaluate the group formation time by measuring the time intervals shown in Figure 1. Figures 2(a) and 2(b) depict



(a) (b) (c)

Figure 3: Fitted lognormal mean (a) and variance (b) values throughout number of devices and the created model (c).

that the average formation times and the standard deviation increase as the number of devices increases for all three schemes. This is because in order to form a group with a higher number of devices, more connections need to be established.

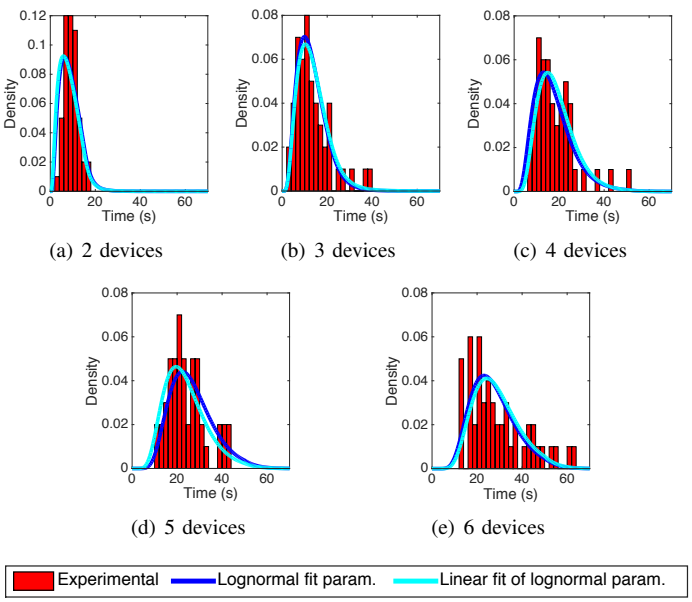
Even though for two devices the average net group reformation time is more or less the same across all the schemes, the *Backup* mechanism has the highest GR_{net} as the number of devices increases due to the burden introduced by the socket messaging. The rate of increase of the *ID-Based* protocol is higher than that of *Random*. The reason is that in the *ID-Based* implementation, the group owner sends a connection invitation to its peers all of a sudden, which creates congestion during the negotiation phase between the group owner and its peers, resulting in a delay in finalizing the connections. As shown in Figure 2(a), the *Random* scheme requires the smallest GR_{net} for networks with more than 3 devices.

Figure 2(b) depicts that the average GR_{total} time increases across the scenarios as the number of devices increases. For two nodes, the *Random* scheme has the highest GR_{total} time due to the random waiting time required for the GO selection. Even though the GR_{net} time for the *Random* scheme is smaller than that of the *Backup* scheme, due to the longer time required to select the GO, the GR_{total} time of the *Random* scheme is greater than that of the *Backup* scheme. This is because the rapid GO selection of the *Backup* scheme can compensate for the higher GR_{net} time. Finally, when considering the GR_{total} time, the *ID-Based* mechanism turns out to be the scheme that needs the highest amount of time for group reformation as the network scales, and is subject to the largest variation, because of the requirement to discover all the devices in the vicinity. Therefore, the *ID-Based* scheme does not appear to be practical, which supports a previous study [7].

C. Modeling the Performance of the Random Scheme

Since the *Random* scheme returns the best performance when increasing the number of devices, it is suitable for implementation in large scale networks. In what follows, we present a model to predict the GR_{total} of a network that involves a higher number of devices.

GR_{total} of the *Random* scheme consists of two stochastic parts, GR_{net} and GO detection, which is based on the minimum waiting time. GR_{net} times for our *Random* scheme experiments have lognormal distributions as determined by a



(a) 2 devices (b) 3 devices (c) 4 devices

(d) 5 devices (e) 6 devices

Figure 4: Validation via comparison of PDFs of lognormal fit, modeled fit and experimental data.

Kolmogorov-Smirnov goodness of fit test with 95% confidence level. Thus, GR_{total} time for *Random* is the summation of the minimum of uniform random variables and one lognormal random variable. The PDF of the group reformation time is hence a convolution of the respective distributions. Mean m and variance v of the lognormal fits along with their linear fits are shown in Figures 3(a) and 3(b), respectively. With n being the number of the devices in the network, the linear fits have the form of $y_m = -6.1012 + 4.9005n$ and $y_v = -34.3234 + 21.5555n$.

We validate our model by comparing our experimental data against the distributions generated by the lognormal fit and the modeled fit. Figure 4 shows this validation applied to networks comprised of two to six devices. The GR_{total} estimations as a function of the number of devices presented in Figure 3(c) further validate our model matching well with the experimental data, with an RMSE of 0.7246. Thus, GR_{total} time is estimated to increase linearly proportional to the number of devices.

Finally, we note that while the evaluation presented in this section involves only 2013 Nexus 7 devices, similar conclusions can be drawn when changing to a different device. In particular, running the experiments presented above on a 2012 Nexus 7, which has an entirely different hardware configuration, returns similar relative results. These suggest that similar conclusions can be drawn for different Android devices [10].

V. Conclusions

In this paper, we propose and explore three practical methods for allowing automatic WiFi Direct group creation and maintenance, and evaluate their performances using commercially available Android devices. For all the proposed methods, we measured the GO selection time and the total group reformation time, to evaluate how quickly these protocols can adapt to connectivity changes.

Our experimental results show that an ID-based group maintenance scheme may not be practical for large networks due to the long peer discovery times. Selecting a backup device ahead of time makes the GO selection almost instantaneous, but it may not be beneficial in highly dynamic networks. Thanks to its small computational burden on the devices and adaptability to an increasing group size, a random GO selection scheme represents the most viable solution, especially with the addition of the step-back mechanism.

Future work includes developing an extension to multi-group networks and the inclusion of an adaptive routing protocol.

References

[1] M. Conti, F. Delmastro, G. Minutiello, and R. Paris, "Experimenting opportunistic networks with WiFi Direct," in *Wireless Days (WD), 2013 IFIP*. IEEE, 2013, pp. 1–6.

[2] A. Pyattaev, K. Johnsson, S. Andreev, and Y. Koucheryavy, "3GPP LTE traffic offloading onto WiFi Direct," in *Proc. of IEEE WCNCW*. IEEE, 2013, pp. 135–140.

[3] W. Alliance, "Wi-Fi peer-to-peer (p2p) technical specification, version 1.5," August 2015.

[4] D. Camps-Mur, A. Garcia-Saavedra, and P. Serrano, "Device-to-device communications with Wi-Fi Direct: overview and experimentation," *IEEE Wireless Communications*, vol. 20, no. 3, pp. 96–104, June 2013.

[5] R. Agarwal, D. Motwani *et al.*, "Survey of clustering algorithms for MANET," *arXiv preprint arXiv:0912.2303*, 2009.

[6] C. Gómez-Calzado, A. Lafuente, M. Larrea, and M. Raynal, "Fault-tolerant leader election in mobile dynamic distributed systems," in *Proc. of IEEEE PRDC*. IEEE, 2013, pp. 78–87.

[7] S. Vaya, "Round complexity of leader election and gossiping in bidirectional radio networks," *Information Processing Letters*, vol. 113, no. 9, pp. 307–312, 2013.

[8] D. R. Kowalski and A. Pelc, "Leader election in ad hoc radio networks: A keen ear helps," *Journal of Computer and System Sciences*, vol. 79, no. 7, pp. 1164–1180, 2013.

[9] B. Tavli, "Protocol architectures for energy efficient real-time data communications in mobile ad hoc networks," Ph.D. dissertation, University of Rochester, 2005.

[10] C. Funai, C. Tapparello, and W. Heinzelman, "Enabling multi-hop ad hoc networks through WiFi Direct multi-group networking," in *Proc. of IEEE ICNC*, Jan. 2017.

[11] Y. Duan, C. Borgiattino, C. Casetti, C. F. Chiasserini, P. Giaccone, M. Ricca, F. Malabocchia, and M. Turolla, "Wi-Fi Direct multi-group data dissemination for public safety," in *Prof. of WTC*, Berlin, Germany, Jun. 2014.

[12] C. Casetti, C. F. Chiasserini, L. Curto Pelle, C. Del Valle, Y. Duan, and P. Giaccone, "Content-centric routing in Wi-Fi Direct multi-group networks," *ArXiv e-prints*, Dec. 2014.

[13] C. Funai, C. Tapparello, H. Ba, B. Karaoglu, and W. Heinzelman, "Extending volunteer computing through mobile ad hoc networking," in *Proc. of IEEE GLOBECOM*, Dec 2014, pp. 32–38.

[14] A. Laha, X. Cao, W. Shen, X. Tian, and Y. Cheng, "An energy efficient routing protocol for device-to-device based multihop smartphone networks," in *Proc. of IEEE ICC*. IEEE, 2015, pp. 5448–5453.

[15] Wi-Fi Alliance, P2P Task Group, "Wi-Fi Peer-to-Peer (P2P) Technical Specification, Version 1.2," Dec. 2011.

[16] D. Camps-Mur, A. Garcia-Saavedra, and P. Serrano, "Device-to-device communications with Wi-Fi Direct: overview and experimentation," *IEEE Wireless Commun.*, vol. 20, no. 3, pp. 96–104, Jun. 2013.

[17] S. Kutten, G. Pandurangan, D. Peleg, P. Robinson, and A. Trehan, "On the complexity of universal leader election," *Journal of the ACM (JACM)*, vol. 62, no. 1, p. 7, 2015.