Reducing Delay in Group Reformation in WiFi Direct Networks through Redundancy

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

***Abstract*—The potential benefits of Mobile Ad Hoc Networks (MANETs) have led to the development of many protocols in order to control and optimize such networks. However, the automatic creation and evolution of ad hoc networks has yet to be exploited. 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 and includes different energy management mechanisms. However, the current WiFi Direct implementations require user interaction for setting up and maintaining the connection. In this work, we exploit redundancy to enable the automatic and fast reconfiguration of WiFi Direct networks in response to network dynamics. We propose a proactive solution to unforeseen group owner failures in order to minimize the packet loss and network discontinuity time by setting up a redundant group on a second virtual network interface. Through emulation on Mininet-WiFi, we find that the proposed redundant scheme substantially decreases packet loss, providing almost continuous connectivity among nodes, which cannot be guaranteed through traditional WiFi Direct schemes. This is the first detailed work examining the auspicious potential of using of an additional network interface to support network reformation using WiFi Direct.**

I. Introduction

Mobile Ad Hoc Networks (MANETs) have been attracting much research attention due to the benefit of networked communication without a fixed infrastructure. Besides offering a complementary solution to the ever-increasing data demand in cellular networks [1], MANETs are also essential in challenging environments, such as disaster areas [2] and battle fields during military operations [3]. The lack of infrastructure necessitates managing many actions and performance metrics of mobile devices. Since in emergency situations connectivity is vulnerable to environmental changes and overall dynamics, efficient and highly responsive organization to maintain connectivity is vital.

Several ad-hoc communication standards and protocols have been introduced to support MANETs. The most prominent ones include IEEE 802.11 DCF, IEEE 802.11s, IEEE 802.11z, ZigBee, SMAC, Bluetooth, and WiFi Direct. However, these protocols showed shortcomings in performance such as low data rates, high energy consumption, short communication range, dependency on user input, or limited ability to scale [4]. They also have not been fully explored as viable solutions for enabling the creation and maintenance of large scale ad hoc networks. Providing the same service quality, e.g., high data rate, security, and power management as traditional WiFi [5], WiFi Direct (initially known as WiFi-P2P) is available in most modern mobile devices [6], [7], and has been identified as a promising candidate for communications in MANETs [8]. WiFi Direct networks are organized in groups, where a device named a group owner (GO) acts as a soft access point, whereas every other device (called Group Members (GMs)) is connected to the GO. Since WiFi Direct inherits all the essentials from WiFi, devices that support WiFi but not WiFi Direct (legacy clients) can join WiFi Direct groups as well. Therefore, WiFi Direct enables ad hoc communication among WiFi devices, e.g., smartphones, tablets, laptops, and printers. Nevertheless, without any automation, WiFi Direct's potential is still far from being fully explored.

Having broad availability and traditional WiFi level capabilities, WiFi Direct has led to many research efforts from single hop to multi-hop networks. Single hop works include, for example, detailed descriptions of WiFi Direct and group formation experiments under various settings on real devices [5], [7], and security considerations [9]. Even though the basic WiFi Direct protocol only allows one GO within a group without specifications for how to perform multi-hop communication, multi-hop studies have emerged due to the potential to create large-scale ad hoc networks [1], [3], [10]–[13]. Researchers have also focused on the optimization of GO selection [14] and group reformation schemes [15]. In [14], the authors select the GO that maximizes the bit rate between GO and GMs, but the authors do not address how much time it takes to select the GO and to form a group, which might cause long periods of disconnection. Since ad hoc networks are inherently dynamic, subsequent GO changes are inevitable, so there needs to be a fast response to unforeseen GO movements or failures. The group formation methods proposed in [15] substantially decrease group reformation time down to around 0.5 seconds in networks involving as many as ten nodes. However, it still takes time to reform the group. Considering the emergency and dynamic circumstances in which MANETs may be deployed, continuity of the network communication is critical.

In this paper, we propose a proactive approach to completely eliminate WiFi Direct network reformation time and to reduce packet delay or drops in case of an abrupt GO failure. We implement a redundant GO scheme by enabling virtual wireless interfaces at each node, and use the additional interface to select a redundant GO that maintains separate connections with all the remaining nodes. When the GO in charge disconnects, the redundant GO takes over immediately with a complete connected network at hand, thus providing zero group reformation time and minimizing packet delay or drop. To the best of our knowledge, this work represents the first approach to achieve additional levels of reliability in WiFi Direct networks by maintaining redundant simultaneous groups.

The rest of the paper is organized as follows. In Section II we describe previous efforts related to this work. Then, we explain our scheme in detail in Section III. Section IV provides a brief overview of the tools used for the implementation, namely WiFi Direct, Mininet-wifi and the B.A.T.M.A.N. routing protocol. In Section V we evaluate the performance of our scheme and compare it against existing approaches. Section VI concludes the paper and presents some future research directions.

II. Motivation and Related Work

The GO in a WiFi Direct network is the key component because it is responsible for controlling the communication throughout the network. As a result, selecting the right device to act as GO is essential. Ad hoc networks are used in highly dynamic and challenging environments where access to an infrastructure is limited, if even possible. Thus, quick and automatic GO selection is inherently critical. In a WiFi Direct network, the GO essentially acts as a cluster head. In the literature, there exists many cluster head selection algorithms that aim at optimizing different metrics, such as energy efficiency, connectivity, and low maintenance [16]. However, they might not be applicable to WiFi Direct networks due to the intrinsic differences of the protocol. Furthermore, they are usually not implemented or tested on real devices, and the current WiFi Direct implementations require explicit user interaction to set up a group.

Given the above, in our previous work [4], we presented three different group reformation schemes to maintain connectivity in WiFi Direct networks, and we implemented them on Android tablets. The proposed schemes are called *Backup*, *ID-Based*, and *Random*. In *Backup*, as the name suggests, the GO in charge elects one of the GMs in the network as the backup GO in case of an unforeseen failure. When the current GO disconnects, the backup node declares itself as the GO and reforms the group. In this scheme, even though the GO selection is immediate, the message exchange required to select the backup GO potentially adds overhead when considering large scale networks. Additionally, the time to reform the network is large, even if the GO is selected immediately. *ID-Based* utilizes node IDs such that, depending on the choice, the node with the smallest (or the largest) ID becomes the GO when the current GO disconnects unexpectedly. Then, this device is in charge of reforming the network by inviting the remaining devices to the newly created group. Even though this is a straightforward and an easy-to-implement scheme, since ranking of the IDs requires a scan for all the available nearby peers for a certain period of time, this scheme is not very promising as the network scales. Furthermore, due to

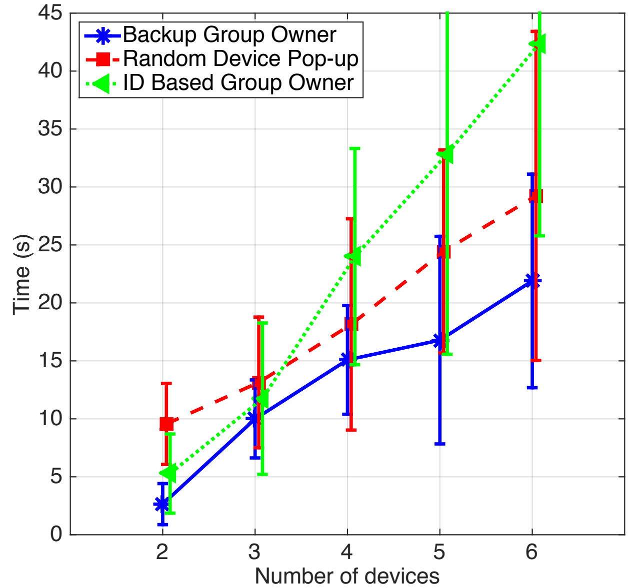

Figure 1: Mean and standard deviation values of the total group reformation times over number of devices for previously proposed GO selection schemes implemented using Android tablets.

interference and channel impairments that affect wireless communications, our experimental results showed that sometimes tablets were unable to detect their rankings, which prolonged the GO selection time or resulted in scenarios were none of the devices elected themselves as the next GO [4]. Better and more promising performance was observed for the *Random* scheme, in which every node starts a uniformly distributed random timer, whose upper and lower bounds are selected such that the probability of simultaneous GO declarations is fixed. When the timer goes off, the node scans for an available GO in its vicinity. If a GO is not detected, then the node declares itself as a GO and starts reforming the network. As shown in Figure 1, all the schemes described here require a non negligible amount of time to reform the network, with reformation times that increase linearly with the number of devices in the network.

Even though these GO selection schemes have been shown to work in practice, they take considerable amount of time to reform the group, partly due to the GO selection scheme and partly due to the long group reformation time. Disconnection periods in MANETs need to be minimized so that the data flow within the network is uninterrupted when a GO leaves. Only if such continuous communication is maintained can MANETs be fully utilized in emergency situations, which is one of their primary use cases.

Previously, a number of studies have been conducted to realize continuous connection with minimal data loss in wireless networks through multiple virtual interfaces [17], [18]. The authors in [17] propose a method to enable WiFi cards to connect simultaneously to multiple APs by creating multiple interfaces, whereas the work presented in [18] utilizes wireless

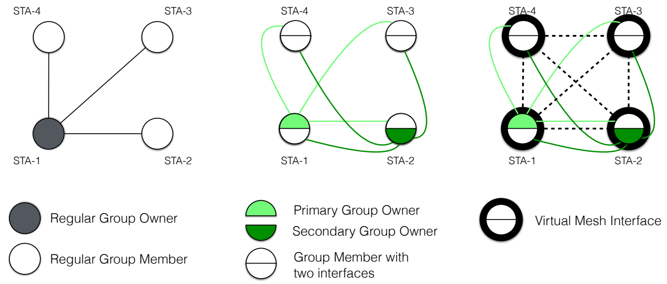

Figure 2: Configurations of a regular WiFi Direct group (left), a WiFi Direct group utilizing the proposed redundant backup group owner approach (middle), and a virtual mesh interface (right).

card virtualization to selectively connect to multiple APs and to opportunistically switch between APs so that continuous TCP data flow is achieved.

By exploiting the ability to create multiple virtual interfaces on a WiFi device, in next section we propose an alternative WiFi Direct group reformation scheme that eliminates the time required for handshakes among nodes, making the GO transition seamless and providing almost continuous data flow within the network. We call this method Redundant Backup Group Owner.

III. Redundant Backup Group Owner

The GO selection schemes presented in [4] are able to reform a WiFi Direct group in the case of unexpected GO failure. However, they have a few shortcomings, such as overhead in large scale networks, potential GO collisions, and most importantly, long group reformation times (see Figure 1). Moreover, in real life applications, the variability of the wireless channel further extends the time it takes to reform a WiFi Direct group. To this end, we propose here a novel proactive scheme we call *Redundant* that aims to reduce the delay in group reformation though redundancy. To do so, we make use of a second interface within the nodes in the network, as shown in Figure 2. Consequently, there are two simultaneous, but separate WiFi Direct groups. We call the two GOs the primary GO and secondary GO for convenience. This does not imply any hierarchy. Instead, the primary GO always refers to the one being used before disconnection. The secondary GO always refers to the one that takes over after the primary GO disconnects. A third virtual interface is added to each of the nodes creating a virtualized mesh network across both sets of WiFi Direct groups. All application traffic is sent and received using this virtual mesh interface.

If the primary GO disconnects abruptly, then the secondary GO takes over immediately, eliminating any period of disconnection, therefore, minimizing the packet delay or reducing any dropped packets.

However, implementation of the *Redundant* scheme is limited in commercially available Android devices. Utilizing multiple wireless network interfaces in Android devices is currently not allowed. Android devices currently do not allow users to create custom virtual wireless network interfaces for WiFi Direct. Nevertheless, there are ways to deal with this [3]. First, by using the WiFi interface for one connection and exploiting the ability of WiFi Direct to support legacy clients, the same device can be connected to multiple groups by mixing the two networks (WiFi legacy client and WiFi Direct). Second, the use of multiple wireless interfaces can be introduced into Android by modifying the operating system so that one can realize the potential of the *Redundant* scheme in real devices. Another issue to examine is how many redundant GOs should exist within a WiFi Direct network for optimized performance in the long run. A complete analysis of the optimal number of redundant GOs as well as the implementation of the *Redundant* scheme in real devices are considered as future work.

IV. Implementation

In this section, we present the details behind the implementation of our proposed *Redundant* scheme. Briefly, the connection among the nodes is realized though WiFi Direct [19]. For routing, we use the B.A.T.M.A.N. (Better Approach To Mobile Adhoc Networking) protocol [20]. We implement the *Redundant* scheme as well as the *Random*, *Backup*, and *ID-Based* schemes in Mininet-wifi [21]. In the following subsections, we provide more detailed information about each of these tools.

A. WiFi Direct

WiFi Direct [19] is an ad hoc communication standard that enables neighboring devices to communicate among themselves without any fixed Access Point (AP). Released by the WiFi alliance, WiFi Direct utilizes IEEE 802.11 a/b/g/n infrastructure mode, and can transmit either at 2.4 GHz or 5 GHz.

Devices need to form a group in order to start communicating with each other. To form a group, one device undertakes the leading role so that it can manage the communication within its group. In WiFi Direct networks, the group leaders are called Group Owners (GO). Every other member of the group is instead called a Group Member (GM). In addition, standard IEEE 802.11 nodes that do not support WiFi Direct can also participate in WiFi Direct groups and are called Legacy Clients (LC). In addition, standard IEEE 802.11 allows nodes that do not support WiFi Direct to participate in WiFi Direct groups as Legacy Clients (LC) It is worth noting that once the group is established, the roles are maintained for the entire duration of the group.

Acting as a soft AP, the GO provides basic service set (BSS) functionalities to the associated clients [19]. The GO has a key role in the functionality of its group, providing responses to incoming join request messages, advertising the group and maintaining the group. Advertisement and group maintainance are handled through beacon packets, just as with a standard IEEE 802.11 AP. Moreover, the GO provides control of the channel to its clients and routes packets being exchanged within the group. Therefore, there is a $1:n$ hierarchy in the group, in which all the GMs and LCs are connected to a single GO.

WiFi Direct devices are capable of utilizing multiple physical or virtual MAC entities. This feature allows them to operate concurrently with a traditional wireless network. Additionally, the specification [19] does not prevent a WiFi Direct device from simultaneously operating as a member of more than one group. Nevertheless, neither multiple MAC functionalities nor simultaneous operations in multiple groups are in the scope of the standard.

WiFi Direct nodes go through a group formation process in which the GO and the GM roles are determined. There are three group formation cases: standard, persistent and autonomous [19], [22]. Standard group formation starts with the nodes listening on channels 1, 6, and 11 in the 2.4 GHz band. After the nodes find each other, they negotiate for the GO role. This is performed through a handshake process, in which the devices exchange an *intent value*, and the one bidding the highest value becomes the GO. After the roles are set, the devices go through a WiFi Protected Setup (WPS) Provision phase. Then, the GO assigns an IP address to the GM using the Dynamic Host Configuration Protocol (DHCP).

The persistent group formation process represents a faster way to recreate a previously established group. In this case, the GO negotiation phase is shortened to the invitation exchange, and exploiting the stored network credentials from the previous connection reduces the WPS Provisioning process considerably. In the autonomous group formation, a node claims the GO role itself and creates its own group.

B. Mininet-wifi Emulator

Mininet-wifi emulator [21] is an open source extension of the pre-existing software defined radio network emulator Mininet [23]. Mininet-wifi enables users to emulate a wide range of realistic and flexible wireless channels and experiments having the support of 802.11 through *SoftMAC* [24]. The Mininet-wifi implementations consist of several elements that act in kernel space and user space. In kernel space, *mac80211_hwsim* is in charge of virtual WiFi interface creation, through which stations (mobile nodes) and APs (static nodes) are created. Mininet-wifi uses a number of standard Linux utilities, such as *iw*, *iwconfig*, which are used for interface configuration and accessing wireless interface information, respectively, and *wpa_supplicant*, which implements the functionality required to support *WiFi Protected Access* through *hostapd*. Mininet-wifi also makes use of the user space utility *Traffic Control*, which configures a Linux kernel packet scheduler to mimic the actual packet behavior of real devices.

Besides these utilities adapted from Linux and 802.11 to conduct emulations that are as realistic as possible, Mininet-wifi includes several mobility and channel propagation models, which do not require any kernel modification. To control the devices' positions and movements, users can determine position, speed, and path of the nodes. The free space channel propagation model is selected by default if no other propagation model is set. If the user wants to implement a new propagation model, it can be obtained though extending a single class. The propagation models introduced in Mininet-wifi provide realistic power and packet loss calculations. As future work, developers plan to improve their models by deploying *wmediumd* and *minstrel*, which enhance the wireless medium for *mac80211_hwsim* and a *mac80211* rate control algorithm, respectively.

Mininet-wifi supports a wide range of network scenarios. Nodes can connect to the APs or form ad hoc connections. Network conditions can be replicated if desired. Vehicular ad hoc networks as well as WiFi Direct are also available for realistic experimentation. Moreover, Mininet-wifi has the ability to run hybrid emulations where virtual devices can be connected to real physical devices, e.g., wireless radios or smartphones.

C. B.A.T.M.A.N. Routing Protocol

Proposed and developed by the Freifunk community [20], the B.A.T.M.A.N. (Better Approach To Mobile Adhoc Networking) routing protocol is the outcome of efforts to address the shortcomings of the OLSR routing protocol, i.e., frequent link disconnection in large networks due to routing loops [25]. Nodes in the network periodically broadcast originator messages in order to let their neighbors know about their existence, and these messages are then used to construct the nodes' routing tables. Each originator message consists of three essential items: the originator address, the address of the relaying node, and a unique sequence number. Upon receiving the originator messages from the neighbors, each node changes the sending address to its address, and then broadcasts the message to its neighbors. This ensures nodes that there are bi-directional links between themselves and their neighbors. Nodes then check the unique sequence numbers to determine where the originator messages come from, hence determining the best route to deliver a packet. Nodes do not store the entire route to the destination. Rather, they keep information for the next link towards the destination. B.A.T.M.A.N. is essentially a mesh routing protocol with OpenFlow compatibility, which means it supports virtualization of multi interface nodes in WiFi cards [26]. The B.A.T.M.A.N. protocol can be tuned by changing the broadcast frequency and time to live of the originator messages and the window size, whose default values are $1second$, 50, and 100, respectively.

V. Performance Evaluation

We tested our *Redundant* scheme in the Mininet-wifi emulator (version $2.2.0d1$) running on Ubuntu version 16.04 LTS. We first manually set up the networks with redundant group owners, including a primary GO and a secondary GO. We emulated our scheme utilizing the free space propagation model without interference. This allows us to directly compare the different schemes without considering the additional effects introduced by the wireless channel. The nodes are located on a grid with $1\ meter$ distance between two neighboring nodes.

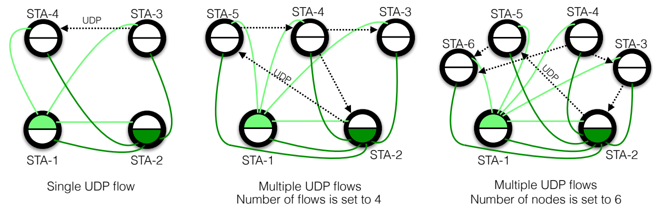

Figure 3: Example UDP flow scenarios between virtual mesh interfaces.

Table I: Mean and standard deviation values of group reformation (GR) times (in seconds) versus number of nodes for previously proposed GO selection schemes implemented in Mininet-wifi.

	Node amount	4	5	6	7	8
Backup	$\mu_{\text{GR time}}$	12.23	15.17	18.07	23.00	25.81
	$\sigma_{\text{GR time}}$	3.71	4.37	4.12	6.43	7.63
ID-Based	$\mu_{\text{GR time}}$	16.98	21.13	24.91	29.41	34.12
	$\sigma_{\text{GR time}}$	6.19	7.21	8.51	8.49	9.47
Random	$\mu_{\text{GR time}}$	18.72	23.61	26.66	30.17	34.98
	$\sigma_{\text{GR time}}$	7.13	7.92	8.67	9.79	10.03

We measured the performance of our scheme through repetitive UDP connections, in which, a total of 5351 datagrams are sent with a bandwidth of $1.05 Mbits/s$ for 60 seconds, where each datagram size is 1470 bytes. We recorded the ratio of dropped datagrams to total datagrams sent when we abruptly disconnect the primary GO.

The main evaluation variables are the number of nodes and the number of UDP flows within the network, where we fix one of the variables and vary the other. First, we focus on scenarios with a single UDP flow selected at random. We excluded the primary GO from the flow selection since it will eventually be disconnected. We also chose flows such that when the primary GO disconnected the virtual mesh interface would be forced to chose a new route through the secondary GO. Otherwise, there would be no datagram loss. The dropped datagram percentages are evaluated based on changing the number of nodes. Second, we increased the number of UDP flows to four, again selecting the servers and clients randomly excluding the primary GO, and evaluating the performance based on the number of nodes. Third, we set the number of nodes to six and evaluated the performance of our schemes as we increase the number of UDP flows in a random fashion. Example test scenarios are provided in Figure 3.

In addition to implementing the *Redundant* scheme presented in Section III, we implemented the previously proposed GO selection schemes in Mininet-wifi in order to determine the performance of the proposed *Redundant* scheme compared to the existing techniques of *Random*, *Backup*, and *ID-Based*, as described previously [4].

We note that the number of dropped UDP datagrams for the previously proposed GO selection schemes is affected by two main components. The first factor is whether the UDP flow is between the new GO and a GM or between two GMs (from our experiments, GO-GM datagram flows tend to recover faster). The second factor is the order of connection, i.e., the order of connection is important, as if the stations that have a UDP flow connect to the GO last, then more datagrams will be dropped. However, we took care of such cases by averaging over 50 realizations. Mean and standard deviations of all of the outcomes of our experiments are displayed in Figures 4-6. In the following, we present detailed outcomes of our experiments.

A. Single UDP Flow

Results for the single UDP flow case are provided in Figure 4. For the previously proposed GO schemes, the percentage of dropped UDP datagrams increases linearly with the number of nodes. This is because as the number of nodes increases, the group reformation time in the GO selection schemes also increases (see Table I where schemes are ordered from fastest to slowest as *Backup*, *ID-Based*, and *Random*). This is because in *Backup*, the GO is pre-determined, so as soon as the current GO disconnects, the backup GO promptly takes over and reforms the network; whereas in *ID-Based*, each node needs to make sure that it has the lowest ID (or highest) among all available peers. This puts the *ID-Based* scheme in the second ranking. On the other hand, in the *Random* scheme, every node picks a random timer as a waiting period before declaring itself as the GO. Due to the fact that the average waiting time increases in order to keep the GO collision probability at 20%, the *Random* scheme takes the longest time to reform the network. With reference to Figure 4, one might notice that the total group reformation times for *Random* approach those of *ID-Based*. This is because, as the network scales, *ID-Based* requires a scan for all available nodes which results in a linear time increase in the group owner selection time. However, the theoretical GO selection time for the *Random* scheme does not increase as fast ($E[min(X_1, X_2, ..., X_n)] = (b+na)/(n+1)$, where X_i are i.i.d. uniform random variables and a, b are upper and lower bounds of the distribution, respectively). Thus, as the network scales, we expect the *Random* scheme to outperform the *ID-Based* approach (for additional details, we refer an interested reader to [4]).

We do not take interference into account in order to explore our schemes to the fullest extent without any disruption from other parameters. Thus, handshakes between nodes take more or less the same time having the mean and the standard deviation values of $3.74s$ and $0.32s$, respectively. Means and standard deviations of group reformation times for the previously proposed group owner selection schemes are provided in Table I. The percentage of dropped datagrams is highly correlated with the group reformation time, even though routing can also contribute by adding additional delay. However, our results for the *Redundant* scheme suggest that routing only plays a minor role compared to the group reformation time.

For the *Redundant* scheme, the UDP datagrams flow

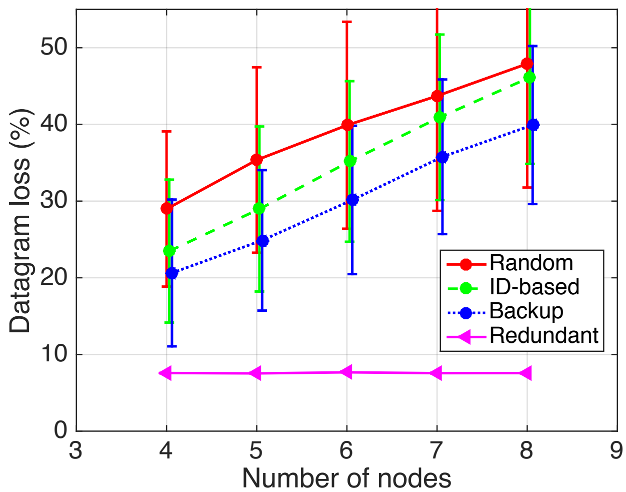

Figure 4: Mean and standard deviation values of dropped datagram percentages in randomly selected and repetitive single UDP flows as the number of nodes in the network is varied.

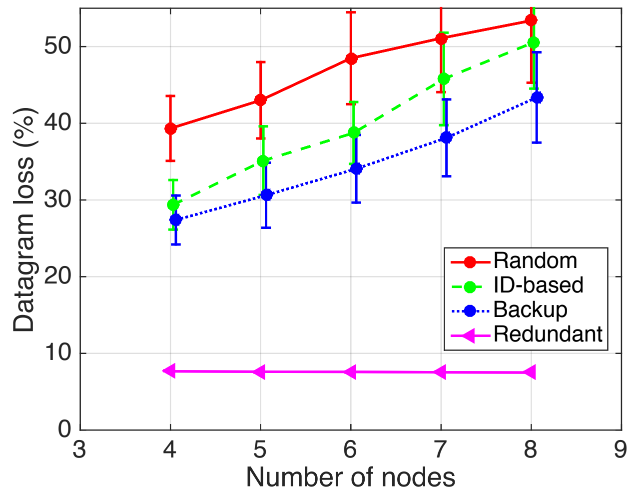

Figure 5: Mean and standard deviation values of dropped datagram percentages for randomly selected and repetitive multiple UDP flows as the number of nodes in the network is varied. The number of UDP flows is set to four.

through the primary GO first. After the node that has the primary GO functionality disconnects abruptly, B.A.T.M.A.N. switches the UDP flows through the secondary GO, which already has all of the connections established. Since the new GO assignment and group reformation is immediate, all the datagram drops are only due to the time required to update the routes. Thus, the dropped datagram rates in the *Redundant* scheme are much lower compared to the previously proposed GO selection schemes. With a mean value of 7.58%, the dropped datagram ratio stays constant as the number of nodes increases. The standard deviations are less than 1%, so we did not display these values on the figure.

B. Multiple UDP Flows - Constant Number of Flows

The evaluation for this scenario was conducted by randomly selecting four UDP flows among the nodes in the network, excluding the GO in charge, which was removed during the ongoing UDP datagram flows. Percentages of dropped datagrams were examined as the number of nodes in the network is increased from four to eight. As shown in Figure 5, the results for the previously proposed GO selection schemes follow the same trends as in Section V-A. The standard deviations in this scenario tend to be smaller than the counterparts in the single UDP flow case, because here we are averaging over the performance of the four UDP flows. This narrows the range of dropped datagram variation. Having a mean dropped datagram ratio of 7.58%, the *Redundant* scheme outperforms all other schemes again with a standard deviation less than 1%, as shown in Figure 5. Because each node handles its own routing table, percentage of datagram lost is not dependent on the number of nodes. As the number of UDP flows increases, the number of datagrams dropped during the disconnection increases linearly. Since the total throughput also increases linearly, the percent datagram loss remains constant.

C. Multiple UDP Flows - Constant Number of Nodes

The evaluation for this scenario is similar to the one described in Section V-B, except in this scenario we fixed the number of nodes to six and varied the number of random UDP flows from two to six. As seen in Figure 6, the results for the previously proposed GO selection schemes follow the same trends explained in Section V-A. However, in this scenario, the rate of increase in datagram loss is less than that for the scenario in Section V-B, which means that the increase in the number of parallel UDP flows does not affect the dropped datagram rate as much as the number of nodes in the network. The previously proposed GO selection schemes are outperformed by the *Redundant* scheme, as clearly shown in Figure 6. The mean dropped datagram ratio over the number of flows is again 7.58% with a standard deviation less than 1%. Due to the fact that every node manages its routing table, the ratios of dropped datagrams do not change.

VI. Conclusions

In this paper, we proposed and explored the performance of a redundant GO scheme in order to minimize the disconnection period in a WiFi Direct network in case of unforeseen GO failures. We stress the importance of seamless connectivity in ad hoc networks in dynamic and emergency situations. By utilizing multiple virtual network interfaces, we select a redundant GO that can route communications after the the primary GO disconnects. This not only completely eliminates GO selection and group reformation time, but also minimizes loss in on-going data flows within the network. Our redundant GO scheme outperforms methods involving a single wireless interface in terms of group reformation time and dropped packet ratio.

Future work includes a comprehensive study on the optimal number of redundant GOs to be selected in a network,

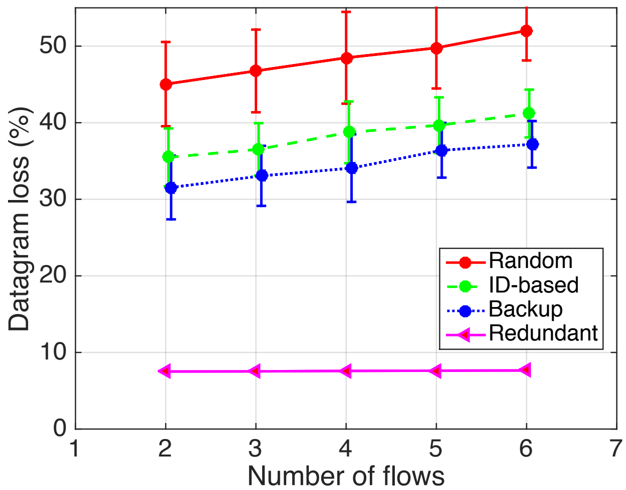

Figure 6: Mean and standard deviation values of dropped datagram percentages for randomly selected and repetitive multiple UDP flows as the number of UDP flows is varied. The number of nodes is set to six.

implementing our multi-interface scheme in real devices, and extending our work to multi-hop WiFi Direct networks.

References

[1] M. Usman, M. R. Asghar, I. S. Ansari, F. Granelli, and K. Qaraqe, "Towards energy efficient multi-hop d2d networks using wifi direct," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, Dec 2017, pp. 1–7.

[2] M. Zhang, M. Yang, Q. Wu, R. Zheng, and J. Zhu, "Smart perception and autonomic optimization: A novel bio-inspired hybrid routing protocol for manets," *Future Generation Computer Systems*, vol. 81, pp. 505 – 513, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X17304545

[3] C. Funai, C. Tapparello, and W. Heinzelman, "Enabling multi-hop ad hoc networks through wifi direct multi-group networking," in *2017 International Conference on Computing, Networking and Communications (ICNC)*, Jan 2017, pp. 491–497.

[4] U. Demir, C. Tapparello, and W. Heinzelman, "Maintaining connectivity in ad hoc networks through wifi direct," in *2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, Oct 2017, pp. 308–312.

[5] 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.

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

[7] 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.

[8] 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.

[9] W. Shen, B. Yin, X. Cao, L. X. Cai, and Y. Cheng, "Secure device-to-device communications over wifi direct," *IEEE Network*, vol. 30, no. 5, pp. 4–9, September 2016.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] K. Jahed, O. Farhat, G. Al-Jurdi, and S. Sharafeddine, "Optimized group owner selection in wifi direct networks," in *2016 24th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, Sept 2016, pp. 1–5.

[15] P. Chaki, M. Yasuda, and N. Fujita, "Seamless group reformation in wifi peer to peer network using dormant backend links," in *Consumer Communications and Networking Conference (CCNC), 2015 12th Annual IEEE*. IEEE, 2015, pp. 773–778.

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

[17] R. Chandra and P. Bahl, "Multinet: Connecting to multiple ieee 802.11 networks using a single wireless card," in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 2. IEEE, 2004, pp. 882–893.

[18] S. Kandula, K. C.-J. Lin, T. Badirkhanli, and D. Katabi, "Fatvap: Aggregating ap backhaul capacity to maximize throughput." in *NSDI*, vol. 8, 2008, pp. 89–104.

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

[20] "BATMAN - Open-Mesh." [Online]. Available: https://www.open-mesh.org/projects/open-mesh/wiki

[21] "The User Manual - Mininet-WiFi," Last time accessed: April 2018. [Online]. Available: https://github.com/ramonfontes/manual-mininet-wifi/raw/master/mininet-wifi-draft-manual.pdf

[22] 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.

[23] R. R. Fontes, S. Afzal, S. H. B. Brito, M. A. S. Santos, and C. E. Rothenberg, "Mininet-wifi: Emulating software-defined wireless networks," in *2015 11th International Conference on Network and Service Management (CNSM)*, Nov 2015, pp. 384–389.

[24] R. d. R. Fontes, M. Mahfoudi, W. Dabbous, T. Turletti, and C. Rothenberg, "How far can we go? towards realistic software-defined wireless networking experiments," *The Computer Journal*, vol. 60, no. 10, pp. 1458–1471, 2017.

[25] D. Johnson, N. Ntlatlapa, and C. Aichele, "Simple pragmatic approach to mesh routing using batman," 2008.

[26] I. T. Haque and N. Abu-Ghazaleh, "Wireless software defined networking: A survey and taxonomy," *IEEE Communications Surveys Tutorials*, vol. 18, no. 4, pp. 2713–2737, Fourthquarter 2016.