Efficient Multi-Group Formation and Communication Protocol for Wi-Fi Direct

Ahmed A. Shahin, and Mohamed Younis
Department of Computer Science and Electrical Engineering
University of Maryland Baltimore County
Baltimore, MD 21250 USA
ashahin1, younis@umbc.edu

***Abstract*—Wi-Fi Direct (WFD) has become almost a standard for peer-to-peer networking on smart devices. Typically, WFD allows a user to search for and connect to other devices, which results in creating a group of devices that can exchange data. In certain applications like alert dissemination in disaster areas, it is required to form groups dynamically and to share data across the group boundaries. However, these capabilities are not offered by the current WFD implementations. In this paper, we propose EMC, a novel protocol that allows WFD devices to dynamically cluster themselves into groups and appropriately elect group owners. EMC also allows members of distinct groups to communicate. EMC is validated through implementation on Android devices and through the realization of a chat application.**

I. INTRODUCTION

Smart devices like iPhone and Android powered phones and tablets have become popular commodity in recent years. These devices support multiple communication standards that enable Device-to-Device (D2D) data exchange at an increased level of convenience. By using technologies such as Bluetooth, Wi-Fi ad-hoc mode and WFD [1], these devices are able to communicate without the need for any communication infrastructures. WFD is distinct among these technologies since it allows both long-range and high-speed communication. Such a feature renders WFD a prime choice for D2D communication in applications such as emergencies, and vehicular networks. In these applications, devices form groups based on interest with the possibility of having multi-group communications. Existing support for multi-groups in WFD is quite lacking. First, there is no protocol for selecting a group leader (owner) among multiple contenders based on certain qualification. Second, although the WFD specifications do not prohibit a certain device from being a member in multiple groups, no guidelines are provided to enable implementation.

In this paper, we propose an Efficient Multi-group formation and Communication (EMC) protocol for WFD. ECM exploits the battery specifications of the devices to qualify potential group owners and enable dynamic formation of efficient groups. The service discovery feature of WFD is utilized to allow devices to share their battery information. A device with a richer energy reserve than those in its range opts for creating a WFD group. Once a group is created, the group owner (GO) uses a service discovery record to distribute its credentials to nearby devices. A device that decides to be a group member (GM) should select one of the nearby GOs to join. Once a group is formed, the GO designates from its GMs what we refer to as proxy members (PMs) that link the group to other groups. Each PM uses its "WLAN" interface to join the group instructed by its GO. To avoid depleting the batteries of the GOs and to adapt for changes in the groups, a teardown signal is sent to notify the devices about restarting the EMC protocol. A typical topology for the network after running EMC is shown in Figure 1. We have chosen the Android platform to implement EMC, due to its popularity, its wide adoption, and the possibility to modify its source code. Part of our implementation of EMC in Android involves the modification of the source code to allow multi-group support by assigning different subnets to each group. A chat application is developed to validate EMC.

In the next section, we compare our protocol to related work. Section III discusses the assumed system model and the problem statement. In section IV, we describe our proposed EMC protocol. The implementation and validation of EMC on real devices are discussed in Section V. Finally, the paper is concluded in section VI with a summary and an outline of planned future extensions.

II. RELATED WORK

Motta and Pasquale [2] were among the first to point out the potential of implementing mobile P2P systems using WFD. They suggested applications that could benefit from WFD such as text messaging, dissemination of traffic information, etc. Such a study has motivated other researchers, e.g., [3][4], to explore the applicability of WFD. Follow up work has introduced protocols that allow the devices in a WFD group to do true P2P data sharing [5][6][7]. The focus has then been

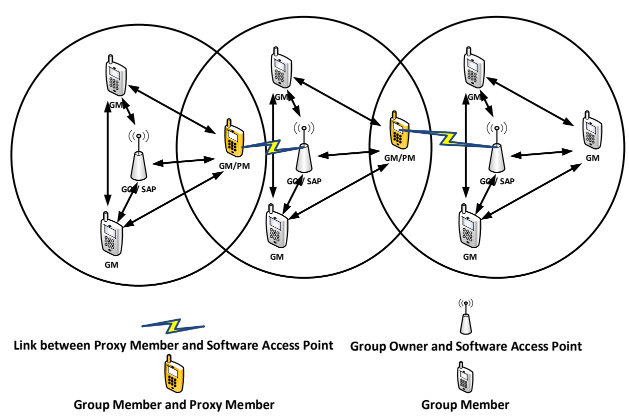

Figure 1: A typical topology for network after running EMC

shifted from intra-group to inter-group data sharing [8]. In addition, the use of the WFD service discovery protocol as a mean of data exchange has been explored in [9][10]. The following highlights how EMC differs from such prior work:

1) The applicability of WFD: Conti et al. [3] used six devices to study the latency in forming a group at the link layer in order to implement an opportunistic networking scheme over WFD. Their work is considered an extension to Camps-Mur et al. [4] who performed real experiments using only two devices. Such a study confirmed the suitability of WFD for P2P systems, and provided measurements of how fast devices form a group.

2) Intra-Group Data Sharing in WFD: In [5] we have introduced a framework that enables devices in one WFD group to communicate by managing the topology changes in the group as well as the data exchange between the devices. The same objective is targeted by Lombera et al. [6], where they proposed iTrust, a peer management over WFD that enable peers to form a mobile ad-hoc network for decentralized information sharing. In addition, Park et al. [7] proposed DirectSpace, a framework for collaboration between devices, which provides a mean for sharing workspaces between users over WFD.

3) Inter-Group Data Sharing: Duan et al. [8] were the first to propose a method for establishing multi-group communication in WFD. They connected two groups together by letting the GO from the first group to connect as a legacy client in the second group using the "WLAN" interface. The authors then experimented with three groups and showed how to connect them together using a combination of unicast and multicast communications. However, their choice for having the GOs to perform the legacy connection imposes certain limitation on how far the GOs should be from each other and the number of groups that a member of a group could reach directly. In addition, they did not propose a dynamic way to create groups and automatically connect them.

4) Service Discovery as a Mean for Data Sharing: In an attempt to overcome the multi-group communication limitation in WFD, we have proposed ADS [9]. ADS is a protocol that uses service discovery in WFD to enable the devices to disseminate alerts without the need for setting up any groups. Wong et al. [10] also exploited the service discovery in WFD to distribute the credentials (SSID, Key) of the Software Access Point (SAP) created by a GO to nearby devices in order to enable them to connect to such SAP and to bypass the confirmation process.

III. System Model and Approach Overview

A. overview of WFD

WFD enables the formation of groups for D2D data exchange, where one of the devices acts as a Software Access Point (SAP) to the other devices in the group; this device is called the group owner (GO). Devices that support WFD associate with the GO using Wi-Fi Protected Setup (WPS), and become group members (GMs). A legacy Wi-Fi device that does not support WFD can participate in the group by associating with the SAP, given that it knows the WPA2 credentials of the SAP (SSID and Key). The assignment of the GO depends on the group formation mode. There are three modes for creating WFD groups, namely, standard, autonomous, and persistent [4]. In the standard mode, the devices negotiate among themselves to elect a GO. In the autonomous mode, one of the devices creates a group and declares itself as a GO. Finally, in the persistent mode, if the same devices start a group again, the previous GO resumes ownership of that group. WFD also supports an optional service discovery protocol that allows a device to define the services it supports using local service records, and to explore what other devices offer using service discovery requests before attempting a connection.

B. Problem Statement and Approach Overview

Enabling D2D communication is highly demanded in the situations where no infrastructure is available. A cost effective and responsive solution is to use WFD to establish D2D connections using the already available smart devices. However, there is no known mechanism in WFD for electing a GO based on certain criteria. In addition, broad exchange of data requires the support of inter-group data sharing by allowing some devices to be members of multiple groups; a feature that the specification of WFD had left to the implementers. Our EMC approach extends the protocol support of WFD to allow dynamic group creation and multi-group communication by introducing the following features:

1) Support of Initial Data Exhcnage: EMC allows certain information like battery information and SAP credentials to be exchanged between devices before connecting the devices together. We utilize our previous work in [9] to perform such data exchange by the help of service discovery in WFD.

- *Battery Information:* This includes charging state, battery level, and battery capacity. A charging device with a high battery level and a large capacity is a good choice for being a GO. Thus, we employ a ranking criteria based on such information, which is used for electing GOs and deciding groups to join. The rank is calculated as follows:

$$Rank = State \times \alpha + \frac{Level}{100} \times \beta + \frac{Capacity}{4000} \times \gamma$$

Where the *State* is 0 or 1 (when charging), *Level* has a range of [0, 100], and the *Capacity* is divided by 4000 mAh, reflecting an average capacity of a number of commercially available devices. The weighting factors α, β, and γ are set to 0.34, 0.33 and 0.33 respectively.

- *SAP Credentials*: EMC follows a similar approach as in [10] and distribute such SAP credential (SSID, Key) using the service discovery protocol. For security reasons, EMC encrypts the SAP Key before distributing it. Such SAP credentials are used for announcing the existence of GOs and to allow PMs to join these GOs as legacy clients.

2) Support of Intra and Inter Group Communication: We utilized our previously-developed framework in [5] to support intra-group communications. Multi-group communication is enabled by associating a device to a group using its "p2p" interface and to another group as a legacy client using its "WLAN" interface. Unlike the work of Duan et al. in [8], EMC connects the GMs not the GOs to other groups as legacy clients to serve as PMs to other groups. This approach allows a group to cover a larger area without constraining the GO placement. Another difference is that EMC creates the groups dynamically which means that the legacy clients selections are not fixed.

IV. Delailed EMC Protocol

The EMC protocol consists of several stages. The first stage is to choose GOs among the candidate devices. Next, the picked GOs create the groups and distribute the SAPs' credentials. In the third stage, the remaining devices choose which groups they should join. Then, the GOs designate PMs from their GMs. Finally, to balance energy consumption, the GOs send teardown messages to inform all devices about restarting the protocol. The transition among the EMC stages is illustrated in Figure 2.

A) Choosing Candidate GOs: Each device creates a local service record that contains its battery information mentioned in the previous section. During this stage, which lasts for a period of T_1, the devices exchange such records by sending service discovery requests. It is envisioned to select T_1 based on the devices density.

B) Creating Groups: After the T_1 period, the devices calculates the *rank* of each device they heard from as well as their own *rank*. A candidate GO, which is the device that has the highest rank among all devices in its range, creates a group in the autonomous mode. Candidate GOs then store the SAP credential of their groups in local service records (SAP records). A device with lower rank declares itself as a GM.

C) Selecting a Group to Join: Each non-GO device stays for a period of T_2 asking nearby GOs, using service discovery requests, for SAP records. The T_2 period is reset once an unseen SAP record is received to allow the device to make better decisions. Once the T_2 period is elapsed, each non-GO device selects a GO to join its group as a GM. The devices does not have to start EMC at the same time. A device that starts the protocol and finds out that a GO exists will go directly to the stage of selecting a group to avoid wasting time in deciding whether to become a GO or not.

D) Selecting Proxy Members: After a GM joins a group, it packages all the collected SAP records and sends it to its GO. Each GO stays for a period of T_3 listening to such type of packets. Once T_3 elapses, each GO picks for each SAP it knows a GM that acts as a PM to that SAP's group. An assignment is then sent to each selected GM, which in turn reacts to the GO message by connecting to the indicated SAP as a legacy client. After connecting to SAPs, each PM acts as a repeater that forwards whatever it receives from this group to the other group that it belongs to.

E) Teardown and restart: A GO waits for a period T_4 before tearing down a group and restart EMC by sending teardown messages. This feature enables balancing the energy load to avoid depleting the batteries of the current GOs, and allows devices with better battery specifications to reform the groups. The selection of T_4 is subject to trade-off between having a balanced load on the devices and frequently incurring the group formation overhead. EMC compensates for timing differences and allows the devices to synchronize by making each PM relay the teardown message to the GO of the other group that this PM is connected to it as a legacy client. If a GO receives the teardown message from the PM before its T_4 period ends, it informs its group members about the teardown. Each device receives the teardown message waits a time T_5 to ensure that other groups are informed before processing the teardown. It is worth noting that the sequential teardown of groups makes sense when they do related tasks. The collective teardown is a means to synchronize them and enable the formation of better groups.

V. EMC Implementation

A. Extending WFD Support in Android

Support for WFD has been provided in Android starting from API level 14. The API level 16 supports simultaneous connections to a Wi-Fi Access Point and a WFD group, and implements the service discovery protocol. However, it is not possible for a device to connect to more than one WFD group at the same time. Another issue in Android is that the range of IP addresses assigned to a group falls in the 192.168.49.x/24 range, where the GO IP address is fixed at 192.168.49.1. Thus even if there is a way to allow multi-group membership, the devices in different groups may not be able to reach each other. As Android is an open source software environment, we chose to extend Android in order to overcome these issues. Basically, we modified the source code to allow the devices to have different subnets. A static assignment is made for the devices involved in the validation. In addition, to allow fully autonomous operation, we changed the source code to allow a GO device to accept connection requests without confirmation

B. WFD Multi-Group Chat Applciation for Android

To validate EMC, an Android chat application that targets API level 16 is developed in AndroidStudio. Once the app runs, a device should be able to chat with devices in its group as well as devices in other groups without any manual interaction. Figure 3 shows screenshots of the app while running on two different devices. Although the app works autonomously, we added manual override buttons to allow testing specific parts.

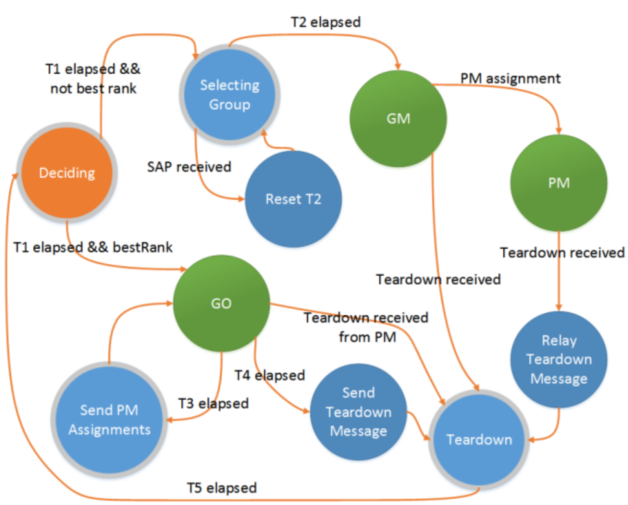

Figure 2: A state diagram summary of the EMC protocol.

C. *Test Cases*

Tests are performed using five devices, two Nexus 4, two Samsung Galaxy Tab 2, and one Asus Transformer tf700t. One of the Nexus 4 devices and the Asus device are kept with their original ROMs with no modifications. The other devices were loaded with our modified version of Android, as mentioned above. By using a Wi-Fi Analyzer app during out tests, we found that all the Wi-Fi channels were overwhelmed, yet EMC had been able to run.

1) Test case 1 -- Group creation: In this test, we used three fully charged devices (one Nexus 4 and Two Samsung Tab). The Nexus 4 device was intentionally plugged in AC source to make its rank the higher among the devices. After running the chat app, the Nexus 4 found that its rank is better than the other devices, so it created a group and the Samsung devices connected to it. The three devices were able to chat together until the predefined time for the teardown, then they restarted the EMC protocol again. The same results were obtained when we ran the app in five devices.

We then tested the incremental running of EMC in the five devices. We started by one of the Samsung tablets, which declared itself as a GO and created a group, since there were no other nearby devices. One of the other devices was then started. Since such a device found an existing group, it skipped the first step of EMC and joined this group, which is the only choice in this case. By repeating the same steps, all the five devices were able to chat together. The teardown mechanism performed well in this case and all of the devices restarted EMC again.

2) Test Case 2 -- Multi-Group Communication: To overcome the device count limitation and space constraints, we added some extra controls in the app to allow a device to bypass the first step in EMC and proceed to creating a group. Starting with five devices, we forced two modified devices (Nexus 4 and Samsung Tab) to create two distnict groups and then tested how the other devices react. The Nexus 4 was kept plugged in the charger, and thus it was the best GO for the remaining devices. The final distribution of the devices was a Nexus 4 as a GO with three other GMs and a Samsung Tab as a GO with no members. All the GM devices in the Nexus 4 group have heard SAP record from the Samsung Tab and knew that another group exists. The three GM devices sent the SAP records for the Samsung Tab to the Nexus 4, which then selected the GM Samsung Tab to serve as a PM. The GM Samsung Tab connected to the SAP of the GO Samsung Tab and became a legacy device in its group. As a result, the devices in the Nexus 4 group were able to chat with the GO Samsung Tab. After finishing the teardown period, the Nexus 4 sent a teardown message that reached the GO Samsung Tab. We noted that all devices responded by restarting the EMC protocol.

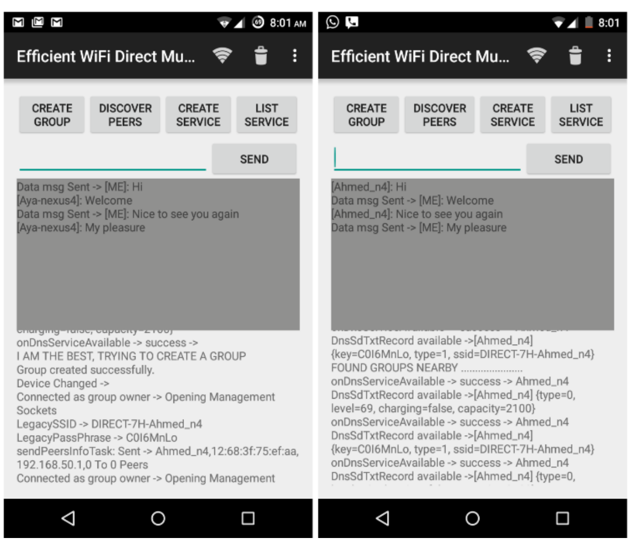

Figure 3: Screen capture of two devices running EMC.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented EMC, a protocol for creating WFD groups dynamically and enabling the application of special criteria for group owner election. EMC takes care also of connecting the created groups together by selecting PMs to relay the data from one group to the others, thus allowing multi-group communications. To avoid depleting the battery of the GOs, EMC restarts itself after a certain period to allow rotation of the GO role. An Android application is developed to validate the EMC protocol. Certain modifications are made to the source code of Android to allow the groups to be in different subnets and to allow the automatic acceptance of connection requests. The applicability of EMC has been confirmed through testing the created application on five smart devices.

In the future, we would like to extend EMC to allow the candidate GOs to negotiate their subnets before attempting to create groups. Another future enhancement is to develop a procedure for selecting PMs to guarantee maximum coverage.

REFERENCES

[1] Wi-Fi Alliance, “P2P Technical Group, Wi-Fi Peer-to-Peer (P2P) Technical Specification v1.2,” December 2011.

[2] R. Motta and J. Pasquale, “Wireless P2P: Problem or opportunity,” *Proc. of the 2nd Int’l Conf. on Advances in P2P Systems*, Florence, Italy, Oct 2010.

[3] M. Conti, F. Delmastro, G. Minutiello, and R. Paris, “Experimenting opportunistic networks with WiFi Direct,” *Proc. of the 6th Wireless Days Conference*, Valencia, Spain, Nov 2013.

[4] D. Camps-Mur, A. Garcia-Saavedra, and P. Serrano, “Device to device communications with wifi direct: overview and experimentation,” *IEEE Wireless Communications Magazine*, Vol. 20, No. 3, pp. 96 – 104, 2013.

[5] A. A. Shahin and M. Younis, “A framework for P2P networking of smart devices using Wi-Fi direct,” Proc. *IEEE 25th Int’l Symp. on Personal, Indoor & Mobile Radio Comm. (PIMRC’14),* Washington, DC, Sep. 2014.

[6] I. Michel Lombera, L. E. Moser, P. M. Melliar-Smith, and Y. T. Chuang, “Peer management for iTrust over Wi-Fi Direct,” *Proc. of the Int’l Symp. on Wireless Personal Multimedia Comm.*, Atlantic City, NJ, Jun. 2013.

[7] Jong-Eun Park, Jongmoon Park, and Myung-Joon Lee, “DirectSpace: A Collaborative Framework for Supporting Group Workspaces over Wi-Fi Direct,” *Proc. of the 4th Int’l Conf. on Mobile, Ubiquitous, and Intelligent Computing (MUSIC 2013),* Gwangju, Korea, Sept 2013.

[8] Y. Duan, et. al, “Wi-Fi Direct Multi-group Data Dissemination for Public Safety,” *Proc. of the World Telecommunications Congress (WTC 2014),* Berlin, Germany, June 2014.

[9] A. A. Shahin and M. Younis, “Alert Dissemination Protocol Using Service Discovery in Wi-Fi Direct,” *Proc. IEEE Int’l Symp. Comm. Software, Services and Multimedia Applications. (ICC’15),* London, UK, June. 2015.

[10] P. Wong, V. Varikota, D. Nguyen, A. Abukmail, “Automatic Android-based Wireless Mesh Networks.” *Informatica* (Slovenia) , 38(4), 2014.