On Designing Energy Efficient Wi-Fi P2P Connections for Internet of Things

Chih-Chiang Liao*, Shin-Ming Cheng*, Menachem Domb†
*Dept. Computer Science and Information Engineering
National Taiwan University of Science and Technology, Taipei, Taiwan
d10515002@mail.ntust.edu.tw, smcheng@mail.ntust.edu.tw
†Dept. Computer Science
Ashkelon Academy, Israel
dombmnc@acad.ash-college.ac.il

***Abstract*—Device-to-Device (D2D) communications enable a wider set of applications and use cases in the Internet of Things (IoT) paradigm. Without the presence of Access Points (APs), Wi-Fi Direct, also known as Wi-Fi Peer-to-Peer (P2P), becomes an intrinsic facilitator for IoT applications due to its popularity. In particular, the P2P group owner (GO) plays the role of the AP and is responsible for making connections with P2P group client (GC). In IoT paradigm, power control should be carefully examined to prevent significant power drop on both Wi-Fi P2P GO and GC. By leveraging the information of connections such as received signal strength and retry count, this paper proposes two novel power control mechanisms for Wi-Fi P2P connections in IoT paradigm. We first design a threshold-based mechanism, which limits the maximum number of connection retries to eliminate unnecessary power consumption. However, considering the dynamic change of wireless condition, an adaptive power control mechanism is proposed to further reduce the waste of energy. We establish an intensive experiment by practically implementing both mechanisms in Wi-Fi P2P devices. The experiment results show that the proposed mechanisms can significantly reduce the power consumption and thus makes Wi-Fi P2P connection more efficient for IoT applications.**

***Index Terms*—Energy Efficiency, Internet of Things, Wi-Fi direct, Wi-Fi P2P**

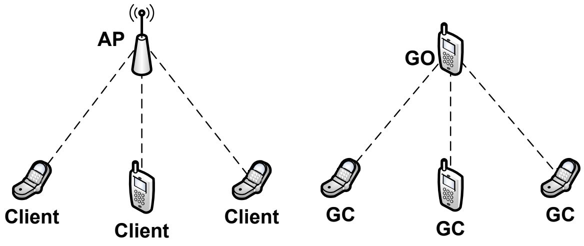
I. INTRODUCTION

With the high-speed transmission rate and easy-to-deploy feature, Wi-Fi has been widely adopted in the human life to fulfill the requirements of various Internet-of-Things (IoT) applications. Typically, Wi-Fi is enabled by deploying infrastructure-based Access Points (APs), which are responsible for providing network connectivity to Wi-Fi clients. To further extend Wi-Fi-based IoT applications, Wi-Fi Alliance (WFA) [1] has recently developed the Wi-Fi Direct technology [2], also known as Wi-Fi Peer-to-Peer (P2P) mode [3]–[5], for the set up of direct device-to-device (D2D) communications among Wi-Fi clients. It is achieved by enabling Wi-Fi P2P devices to negotiate the roles of AP (known as group owner; GO) and client (known as group client; GC), and to establish an infrastructure-like connection. In this case, Wi-Fi P2P mode is beneficial from the existing basis of Wi-Fi infrastructure mode by directly considering Wi-Fi P2P GO as a legacy AP. The related IoT applications on the top of Wi-Fi P2P mode have been developed accordingly [6]–[8].

However, in Wi-Fi P2P mode, both GO and GC might be the portable devices, which implies that the power consumption problem becomes more critical compared with the power sufficient legacy AP. Moreover, Wi-Fi-based IoT applications are particularly power-consuming, and battery capacity in portable device grows much slower, making an energy-efficient design for Wi-Fi P2P as an important issue for IoT. Recently, literature started to concentrate on the power-saving problem for an initiated group by speeding up connection establishment [9], by designing group reformation [10], by adjusting sleep period [11], or by managing group communications [12], [13]. However, no related research focused on the polling-based power control for active but unstable Wi-Fi P2P connections, which tries to resend packets to maintain the group connections no matter the current quality of the connection, thereby causing a busy loop and wasting dramatic energy [3].

By leveraging the information of connections such as received signal strength and retry count, this paper proposes two novel power control mechanisms for Wi-Fi P2P active but unstable connections to prevent the occurrence of the busy loop. We first design a threshold-based mechanism as the baseline, which could limit the maximum number of connection retries to eliminate unnecessary power consumption. In particular, when the received signal is weaker than a predefined threshold, we stop sending the retry packet. Considering the dynamic change of wireless condition, an adaptive power control mechanism is consequently proposed to decrease the negative effects of a busy loop for retries. In this case, the stop condition for retry is adjusted according to the current status of the connection. To achieve backward compatibility, the proposed mechanisms are designed to be integrated into the existing software retry logic protocol seamlessly.

In order to evaluate the performance of the proposed mechanisms, we conduct intensive experiments where both mechanisms are implemented in Android system over Nexus 7. The experimental results show that the adaptive mechanism can significantly decrease the power consumption and thus makes Wi-Fi P2P connection more energy-efficiently. To the best of the authors knowledge, there is no previous work designing connection retry algorithm based on the Wi-Fi P2P specification. The practical implementation for validation



(a) legacy Wi-Fi infrastructrue (b) Wi-Fi P2P

Fig. 1. One-to-many architecture of Wi-Fi networks

makes this work more convincing and proves the compatibility of the proposed mechanism. We believe that the proposed energy-efficient retry mechanism could aid in the widespread deployment of Wi-Fi P2P technology for IoT applications.

The rest of this work is organized as follows. The background and related work are described in Section II. This section also describes the operation of retry for unstable Wi-Fi P2P connections. Section III introduces the system model, and Section IV describes both threshold-based and adaptive retry control mechanisms. In Section V, we conduct realistic experiments to evaluate the performance of the proposed mechanism. Finally, we conclude this work in Section VI.

II. BACKGROUND AND RELATED WORKS

This section describes the background of Wi-Fi P2P and the recent literature related to power control.

A. Wi-Fi P2P Group Formation

To achieve D2D Wi-Fi communications, Wi-Fi P2P devices form a P2P group, where a P2P Group Owner (GO) acts as an AP for a set of connected P2P Group Clients (GCs). Typically, two P2P devices run a 4-way handshake negotiation protocol after discovering each other, where the P2P GO shall act as the authenticator and the P2P GCs shall act as the supplicant [14]. As shown in Fig. 1, the P2P GO may provide a data distribution service between all connected GCs in the P2P Group. The resulting temporal encryption keys shall be installed and used to encrypt unicast and broadcast/multicast frames exchanged between the P2P GO and GCs.

B. Power Control on P2P Mode

Since the formation of a group is time- and power-consuming, it is important to keep the stability of a group. To maintain a valid group, GO will periodically check GC's status so that an unstable GC can be identified. Typically, it is achieved by a classic polling mechanism, as shown in Fig. 2, once GO observes that the failure rate of packets sent to GC (i.e., r_f) is higher than the predefined threshold f_{th}, GO will periodically try to make a request to the GC. The "retry procedure" is implemented by using a busy loop and of course incurs heavy power consumption. For mobile IoT applications, this case becomes more serious because the distance between GO and GC changes frequently. As a result, such retry for group maintainness should be carefully considered to prevent the significant drop in energy.

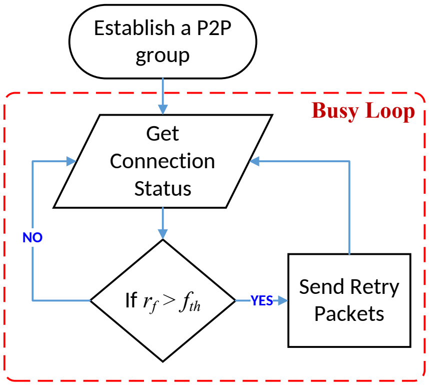


Fig. 2. Polling behavior in P2P group owner

TABLE I
THE CORRESPONDING VALUES OF RSSI, TXGOOD, AND TXBAD FOR EACH SIGNAL LEVEL

Signal Level	4	3	2	1	0
RSSI	-50dBm	-60dBm	-75dBm	-83dBm	-90dBm
TxGood	80-90	63-69	45-48	3-5	ignore
TxBad	5-10	23-26	42-51	70-80	ignore

Chak *et al.* [10] proposed an energy-efficient algorithm to perform Wi-Fi P2P group's reformation when the group is unstable. However, to the best of the authors knowledge, there is no previous work designing connection retry algorithms based on the Wi-Fi P2P specification. In legacy Wi-Fi, AP could leverage the signal strength to reduce unnecessary consumption [15]. Gandarillas *et al.* [16] further exploit transmitting and receiving signals to predict possible power consumption.

III. SYSTEM MODEL

We consider a Wi-Fi P2P group consisting of one GO-GC connection. A classic polling-based mechanism for the maintenance of a P2P group is adopted. In particular, when the failure rate of packets sent to GC is larger than a predefined threshold, Wi-Fi channel saturation, poor device location, or deficient coverage are indicated. To make the mechanism most energy-efficient, we leverage Received Signal Strength Indicator (RSSI), TxGood, and TxBad at GO to estimate the GO-GC connection status so that the unnecessary retries can be minimized. The details of these two factors are described as follows.

- RSSI at GO. It indicates the quality of the connection from GC to GO, which might reflect the quality of the connection from GO to GC of some sort. We leverage this factor due to that it can be easily retrieved from the GO side.
- TxGood and TxBad at GO. The two factors can be exploited as the indicator of the transmission quality. Basically, it is calculated by using the physical layer parameters. If the quality of the transmission is more acceptable, the value of TxGood is higher, and TxBad is smaller.

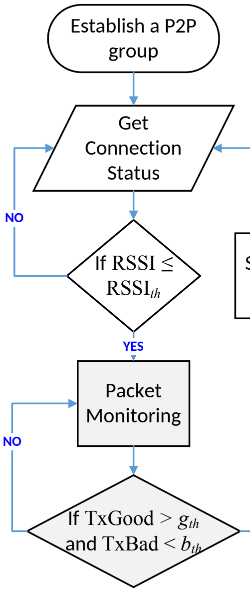


Fig. 3. Opeation logic of threshold-based retry control

This paper primarily considers the power consumption and the number of retries for the GO-GC connection. Moreover, we are interested in the quality of transmission, and we exploit "signal level" as the indicator. Table I summarizes the corresponding values of RSSI, TxGood, and TxBad for each signal level. By investigating the retry count of transmissions with different signal levels, we can understand the effects of elimination on unnecessary retries more clearly.

IV. RETRY CONTROL MECHANISMS

This section proposes two new mechanisms, threshold-based and adaptive, for retry control in polling of Wi-Fi P2P connection to eliminate energy waste due to unnecessary retry. The details are described in the following subsections.

A. *Threshold-based Retry Control*

Fig. 3 shows the operational details of the proposed threshold-based retry control mechanism for Wi-Fi P2P connections. We can easily observe that in the original operation flow, GO blindly tries to make a connection no matter what the current quality of the connection is. In some cases, it is not possible to make a connection since the quality of the connection is very bad. The critical problem is that the "failure rate" is the result of the bad connection while the RSSI is the cause of the bad connection.

Comparing with the operation flow of legacy Wi-Fi P2P try control, the threshold-based mechanism simply exploits RSSI as the parameter to determine if the quality of the P2P connection is bad. Once the RSSI is lower than the threshold RSSI_{th}, instead of entering connection retry, the proposed mechanism passively monitors the quality of the P2P connection by checking TxGood and TxBad. If TxGood is larger than a threshold g_{th} and TxBad is smaller than a threshold b_{th}, it indicates that the condition of P2P connection is recovering, and thus the connection retry becomes more meaningful. In this case, the success possibility of making a reconnection becomes higher and the possibility of unnecessary retry becomes smaller. As a result, an energy-efficient P2P connection is expected.

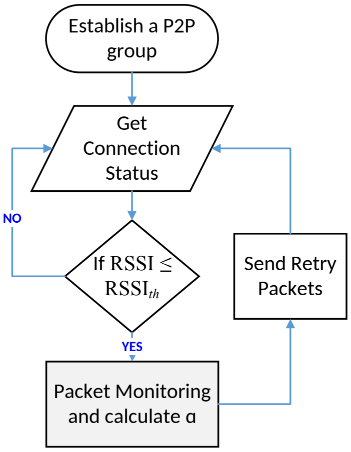


Fig. 4. Operation logic of adaptive retry control

B. *Adaptive Retry Control*

We can observe that the original operation flow of Wi-Fi P2P follows the "always retry" rationale while the threshold-based mechanism follows the "retry when necessary" rule. However, the determination of necessary timing is difficult and needs to be carefully designed. This subsection introduces an adaptive approach where GO decides the number of retries by using the current status of the connection adapters. In particular, we define α to represent the status of the connection as follows.

$$\alpha = \left(\frac{\text{RSSI}}{\text{RSSI}_{\max}}\right) \times \left(\frac{\text{TxBad}}{\text{TxGood} + \text{TxBad}}\right). \tag{1}$$

It is obviously that α considers RSSI, TxGood, and TxBad concurrently. In this case, the number of connection retries performed by GO is limited and significant power consumption due to the unlimited number of retries in the original procedure is prevented. As shown in Fig. 4, when the quality of thge connection is good, the number of retries performed by GO is larger while the quality is bad, GO performs retry procedure less.

V. PERFORMANCE EVALUATION

To validate the correctness of the proposed mechanism and evaluate its performance, we build an intensive experiment where we choose Nexus 7 as the platform for implementation and testing. The protocol stacks of Nexus 7 is clear, and it is easy to insert the codes in such open source system, compile it and then update the binary to the test bed. By adding the debug logs, we can understand the operation flow of retry control in Wi-Fi P2P. As shown in Fig. 5, the experimental environment consists of one GO and one GC, which respectively connect

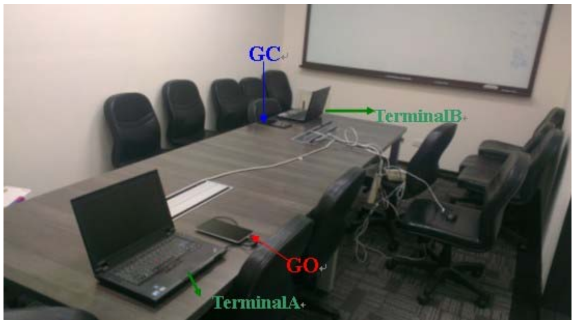


Fig. 5. Experimental environment

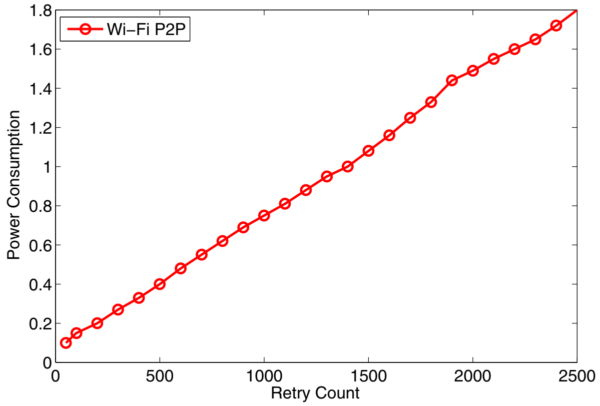


Fig. 6. Power consumption of retry count

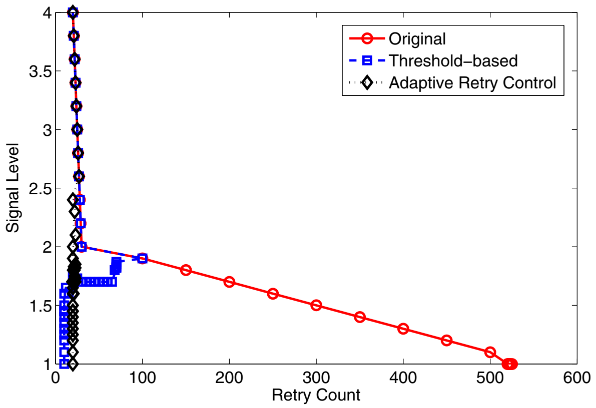


Fig. 7. Signal level of retry count

to terminals for debugging. Each P2P device (i.e., Nexus 7) is equipped with 1.5GHz CPU, 2GB RAM, and 1xUSB interface.

Table II shows the number of retries (known as retry count) of the connections at each signal level. We can simply observe that in the original polling-based retry mechanism, unnecessary retries exist when the status of the connection is poor. By introducing the threshold-based mechanism, the number of unnecessary retries decreases significantly, thereby reducing the power consumption.

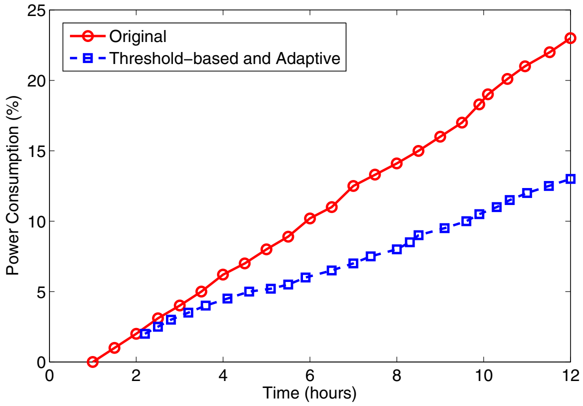


Fig. 8. Power consumption of original, threshold-based, and adaptive retry control mechanisms

In Fig. 6, we observe an intuition result that the power consumption will be in proportion to the number of connection retries. This figure also validates the correctness of the experiment. Fig. 7 shows the relationship between the signal level and retry count. For the original scheme with the polling-based retry scheme, when the signal level is from 4 to 2, the retry count is low and stable. It is due to the reason that with good transmission quality, no heavy retry occurs. As predicted, when the signal level is below to 2, a huge increase in retry count is observed. When the signal level decreases to 1, no retry exists since the group connection is crashed.

We can observe in Fig. 6 that the proposed threshold-based and adaptive schemes perform very well when the quality of transmission is bad (i.e., the signal level is below to 2). It is due to the fact that in the proposed scheme, the unnecessary retries have been stopped when the quality of transmission is poor. This figure also shows that the adaptive retry control performs better than the threshold-based one, for example, the retry process will be stopped even in signal level 3 in adaptive retry control scheme. It is due to the design of α in equation (1).

In Fig. 8, we observe that in the original scheme, the polling-based retry scheme consumes 23 percent of energy totally in 12 hours. With the aid of the proposed scheme, the unnecessary energy waste is eliminated, and the control scheme only consumes 10 percent of energy.

VI. CONCLUSION

For more efficient manage power for unstable Wi-Fi P2P connections for IoT, this work proposes two retry control mechanisms, where we leverage more information of connection to make a smarter decision. Instead of performing a busy loop to achieve a successful retry, we first propose a threshold on the retry counts, so that the power waste due to the unnecessary retry is avoided. The proposed adaptive mechanism could dynamically control the retry according to the current status of connections. To practically validate the proposed mechanism, we conduct realistic experiments by implementing the mechanisms in the existing Android

TABLE II
SIGNAL LEVEL AND RETRY COUNT

Level	4	3	2	1	0
Original	20-30	25-50	55-70	about 500	0=>disconnected, no retry
Threshold	20-30	20-30	25-45	1x =>stop the retry	0=>disconnected, no retry

system over Nexus 7. The experimental results show that the proposed mechanisms significantly decrease the unnecessary power consumption comparing with the legacy polling-based retry scheme. The experiment also proves that the proposed mechanisms are compatible to the existing solution, which therefore facilitates the wide development of Wi-Fi P2P technologies.

ACKNOWLEDGEMENT

This work is supported in part by Ministry of Science and Technology, Taiwan, under contract numbers MOST 105-2218-E-011-003 and 105-2628-E-011-001-MY3.

REFERENCES

[1] Wi-fi alliance. [Online]. Available: http://www.wi-fi.org

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

[3] P2P Technical Group, “Wi-Fi Peer-to-Peer (P2P) Technical Specification v1.0,” Wi-Fi Alliance, Tech. Rep., Dec. 2009.

[4] D. Karvounas, A. Georgakopoulos, K. Tsagkaris, V. Stavroulaki, and P. Demestichas, “Smart management of D2D constructs: An experiment-based approach,” *IEEE Commun. Mag.*, vol. 52, no. 4, pp. 82–89, Apr. 2014.

[5] Y. Chao, H. Zhang, and L. Song, “Demo: WiFi multihop - implementing device-to-device local area networks by Android smartphones,” in *Proc. ACM MobiHoc 2015*, June 2015, pp. 405–406.

[6] P. Chaki, M. Yasuda, and N. Fujita, “A Wi-Fi P2P communication platform between wireless LAN memory cards: Prototype implementation and performance evaluation,” in *Proc. IEEE CCNC 2016*, Jan. 2016.

[7] K. Liu, W. Shen, B. Yin, X. Cao, L. X. Cai, and Y. Cheng, “Development of mobile ad-hoc networks over Wi-Fi Direct with off-the-shelf Android phones,” in *Proc. IEEE ICC 2016*, May 2016.

[8] J. V. Santos, D. V. Torres, M. K. Villanoy, C. A. Festin, J. N. Doctor, and R. Ocampo, “Rapid mobile development with ARC: Application framework for robust communications for disaster risk reduction and management,” in *Proc. IEEE WiMob 2016*, Oct. 2016.

[9] A. A. Shahin and M. Younis, “Alert dissemination protocol using service discovery in Wi-Fi Direct,” in *Proc. IEEE ICC 2015*, June 2015, pp. 7018–7023.

[10] P. Chaki, M. Yasuda, and N. Fujita, “Seamless group reformation in wifi peer to peer network using dormant backend links,” in *Proc. IEEE CCNC 2015*, Jan. 2015, pp. 773–778.

[11] H. Yoo, S. Kim, S. Lee, J.-Y. Hwang, and D. Kim, “Traffic-aware parameter tuning for Wi-Fi Direct power saving,” in *Proc. ICUFN 2014*, July 2014, pp. 479–480.

[12] A. A. Shahin and M. Younis, “A framework for P2P networking of smart devices using Wi-Fi Direct,” in *Proc. IEEE PIMRC 2014*, Sept. 2014, pp. 2082–2087.

[13] G. Z. Khan, R. Gonzalez, E.-C. Park, and X.-W. Wu, “A reliable multicast MAC protocol for Wi-Fi Direct 802.11 networks,” in *Proc. EuCNC 2015*, June 2015, pp. 224–228.

[14] P. T. Group, *Wi-Fi Peer-to-Peer (P2P) Technical Specification*, Wi-Fi Alliance Std., Rev. 1.0, Dec. 2009.

[15] H. Li and L. Chen, “RSSI-aware energy saving for large file downloading on smartphones,” *IEEE Embedded Systems Letters*, vol. 7, no. 2, pp. 63–66, June 2015.

[16] C. Gandarillas, C. Martn-Engeos, H. L. Pombo, and A. G. Marques, “Dynamic transmit-power control for WiFi access points based on wireless link occupancy,” in *Proc. IEEE WCNC 2014*, Apr. 2014, pp. 1093–1098.