Performance Evaluation of WiFi Direct for Data Dissemination in Mobile Social Networks

Zhifei Mao*, Jing Ma†, Yuming Jiang* and Bin Yao†
*Norwegian University of Science and Technology (NTNU), Trondheim, Norway
†Shanghai Jiao Tong University (SJTU), Shanghai, China

***Abstract*—WiFi Direct is a recent device-to-device communication technology standardized by the WiFi Alliance. Its increasing availability on popular mobile systems (e.g. Android) presents a unique opportunity for developers to implement mobile social networks (MSNs), a new paradigm that facilitates data dissemination without Internet access by leveraging human mobility and short-range communication technologies. Since WiFi Direct is not originally designed for such applications, it is significant to learn its performance in practice. In this paper, we investigate goodput and fairness of WiFi Direct for data dissemination in MSNs. To this end, we develop an MSN application and conduct three sets of experiments on a testbed comprising several Android devices. Experimental results show that the data loads and mobility of nodes greatly impact the goodput and fairness.**

I. Introduction

The widespread use of smart mobile devices and the technologies of short-range wireless communications facilitate data communication over mobile social networks (MSNs). By leveraging human mobility and short-range communication technologies on their mobile devices, people in MSNs can share digital content and form chatting groups without Internet access. MSNs can serve as a complementary solution to traditional online social networking sites, which can be used when the Internet is unavailable. It is also a salient technology for mobile data offloading from cellular networks that are overloaded by multimedia contents from e.g. YouTube and for social commerce advertising for small businesses.

WiFi Direct [1], which supports typical WiFi speeds and a transmission range up to $200m$, is a favorable technology for data dissemination in MSNs. WiFi Direct devices connect to each other by forming groups. In a group, one node is selected as group owner (GO) to control the group like a conventional access point (AP), while other nodes connect to the GO as clients. Since WiFi Direct is directly built on traditional WiFi infrastructure mode, it does not require dedicated hardware to support its functionalities. Therefore, it is now natively included in many mobile systems (e.g. Android 4.0 and above). Recently, researchers have demonstrated the feasibility of using WiFi Direct as the medium for multihop networking [2], multigroup networking [3], [4], and opportunistic networking [5], which are the underlying networking techniques for MSNs. Motivated by these positive results, we move forward to investigate the goodput and fairness of WiFi Direct in data dissemination in MSNs. Goodput is crucial in data dissemination in MSNs, as the duration of a contact between nodes is typically short due to their mobility and consequently the amount of data that can be disseminated during the contact is considerably limited. Instead of throughput-based and time-based fairness [6], we look into application layer fairness with respect to data dissemination rate of the nodes, i.e. how fast the data of each node can be disseminated.

To this end, we develop an MSN application that enables data dissemination among a group of WiFi Direct nodes. By its original design, WiFi Direct does not define client to client communication [2]. In our implementation, we allow the clients to upload their data to the GO that later forwards the received data to other clients. In addition, data can be disseminated under two types of transmission strategies in the application. One is the contention-based strategy that directly uses the built-in distributed coordination function (DCF) to schedule the transmissions of the nodes. Since DCF can cause severe collision when the data load is heavy, we implement another cooperation-based strategy where the GO allocates exclusive slots to every node and schedules all the data transmissions on the application layer. In this work, we perform several sets of experiments to investigate the impact of slot size, data load and mobility on the performance of WiFi Direct under these two strategies. The experimental results show the advantage of the cooperation-based strategy with large slot size on achieving high aggregate goodput and its flexibility when facing asymmetry of data load and mobility. To the best of our knowledge, this is the first experimental work to study the fairness of WiFi Direct in data dissemination and goodput performance against mobility.

The rest of this paper is organized as follows. Section II gives a brief background of WiFi Direct and introduces two transmission strategies for data dissemination with WiFi Direct. We present our implementation of the two strategies in Section III. Section IV presents our experimental evaluations. We conclude in Section V.

II. Background

A. WiFi Direct in Brief

WiFi Direct is built on the prominent WiFi infrastructure mode and enables device-to-device communication on WiFi channels [5]. Devices that are WiFi Direct enabled communicate with each other through P2P groups. Within a group, one of the WiFi Direct devices acts as GO to control the group including managing node join/leave, and starting/terminating the group. Other devices in this group connect to the GO as clients. The GO is effectively a soft AP that provides some functionalities of infrastructure AP, such as the basic service set functionality and WiFi Protected Setup. To become the GO, a device has to be WiFi Direct enabled, while the clients can be WiFi Direct devices or normal WiFi devices. For a more detailed overview of WiFi Direct, please refer to [7].

B. Data Dissemination with WiFi Direct

Data dissemination in MSNs exploits opportunistic contacts between mobile nodes. WiFi Direct is a favorable communication technology for such data dissemination due to its long transmission range and high data rate, in comparison to other alternatives such as Bluetooth and NFC.

When a number of MSN nodes with WiFi Direct enabled come into each other's transmission range, they first form a group by following the group formation process of WiFi Direct. Once the group is established, the nodes can disseminate their data to other nodes in the group. WiFi Direct does not define the communication between clients [2], as each client does not know the information of other clients including IDs and IP addresses by its original design. Therefore, one has to implement additional function along with the MSN application to allow the data of all nodes being shared with others. Since the GO is able to communicate with every client directly, we consider a dissemination approach where the data of one client can be first uploaded to the GO and then forwarded to all other clients by the GO[1]. To avoid changing the MAC and network layer functionalities, which may affect the operation of other WiFi Direct based applications, it is preferred that this additional function is implemented on the application layer.

C. Transmission Strategies

Since it is built on top of WiFi, WiFi Direct uses DCF to share the wireless channel among devices in the same group[2]. With DCF, nodes that have data to transmit need to content for channel access, which causes collision and data retransmission. However, the central role of the GO provides possibility for cooperative transmissions on the application layer. This can be realized simply by the GO sending the clients control messages to inform them to start/stop their transmissions, so that one node can obtain a dedicated period to transmit. As a result, the performance such as goodput can be improved by alleviating the channel contention. In our study, we consider both contention-based strategy and cooperation-based strategy for local data dissemination, and investigate their performance in practice. The following provides a description for these strategies:

1) Contention-based strategy (TBS): TBS does not limit the functioning of DCF regarding channel usage on the application layer. All the nodes are free to join in the contention for channel access if they have data to send.

2) Cooperation-based strategy (PBS): The transmissions of all nodes' data to be disseminated are scheduled at the GO. Each node is allocated by the GO dedicated time slots for data transmission. We use round-robin scheduling algorithm to determine the schedule and slot sizes. Detailed description is presented in III-B.

D. Performance Metrics

The focus of our study is to investigate the performance of TBS and PBS in local data dissemination, and seek the cases where PBS can possibly outperform TBS. The performance metrics of interests are

1) Goodput: It measures the amount of data delivered per unit time to the application layer. Individual goodput and aggregate goodput are investigated.

2) Proportional fairness: Define *data dissemination rate* of a given node k the amount of k's data per unit time received by all other nodes in the group. It is said to be fair if the data dissemination rate, R_k^d, of each node k is proportional to its data load M_k. We use Jain's fairness index [10] to measure the achieved fairness level of TBS and PBS, which is defined as

$$FI = \frac{(\sum \frac{R_k^d}{M_k})^2}{n \cdot \sum(\frac{R_k^d}{M_k})^2} \tag{1}$$

where n is the number of contending nodes that have data to disseminate. FI is bounded between 0 and 1; larger FI indicates that the allocation of a dissemination strategy is closer to a proportionally fair allocation.

III. Implementation

We have developed a simple MSN application on Google Nexus 6P that was announced in September 2015. Nexus 6P runs Android 6.0.1 Marshmallow (API level 23) and supports WiFi Direct. The application is built on the Wi-Fi P2P APIs, and it enables messaging and file sharing among a group of WiFi Direct nodes over TCP socket connections.

Since data dissemination in a WiFi Direct group, using no matter TBS or PBS, requires each client to upload its data to the GO that later forwards the received data to other clients, it is sufficient to establish socket connections only between the GO and the clients. Specifically, for each socket connection between the GO and a client, the GO serves as TCP server and runs `ServerSocket`, while the clients run client `Socket`.

A. GO Forwarding

The GO dedicates a queue for each client to store the data uploaded from it. To facilitate data forwarding, the GO also keeps a log file for each client i, which tracks the history of data uploading by i and forwarding by the GO. In particular, a $\mathtt{position(i,j)}$ value is recorded in the log file, which indicates the starting data byte for the coming forwarding to another client j. When the GO is able to use the channel to forward client i's data to client j, it will check $\mathtt{position(i,j)}$ and then send i's data in the queue from the starting byte indicated by $\mathtt{position(i,j)}$. Once the forwarding is finished, $\mathtt{position(i,j)}$ will be updated.

[1] MAC layer broadcast is also possible, however, it is not considered in our study due to its low data rate and unreliability [8].

[2] Point coordination function is another MAC technique used in IEEE 802.11, which allows AP to coordinate the communication within the network, however it is not implemented by the Wi-Fi Alliance in its interoperability standard [9].

B. Transmission Scheduling in PBS

For TBS, whenever a node has data to send, it competes for channel access with other nodes that would like to send data at the same time. For PBS, the transmissions of all the nodes are scheduled at the GO in a round-robin way. Suppose there is a group $N = \{0, 1, 2, ..., n\}$ of nodes where node 0 is the GO, and nodes $i = 1, 2, ..., n$ are clients. Fig. 1 shows the schedule for each round of transmissions. In each round, the GO disseminates its data to all the clients first, then each client uploads its data to the GO that later forwards the received data to other clients.

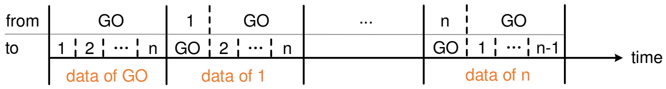


Fig. 1: PBS transmission scheduling.

As can be seen from Fig. 1 that PBS allows each node to transmit data during its exclusive time slots. In addition, time slots are data-centric, namely, during a slot, the channel is used to transmit data of a given node. A slot dedicated to data of node k, denoted by s^k, is composed of n sub-slots. Specifically, the slot for data of the GO has n *sending sub-slots*, where each sending sub-slot is used to send the GO's data to a client. The slot for data of client i has an *uploading sub-slot* and $n-1$ *forwarding sub-slots*. During the uploading sub-slot, client i sends its data to the GO, while during a forwarding sub-slot, the GO forwards the received data to one of other clients. Denote t^k the size of slot s^k. The slot sizes can be determined in various ways. In our study, we let t^k be proportional to the data load of node k and be equally allocated to n sub-slots. The objective is to allow the data of all the nodes to be disseminated almost at the same time. Assume each node k has a volume M_k of data to be disseminated to other nodes in the group. Then we have $t^k = \frac{M_k}{min_j\{M_j\}} \cdot \tau$ where τ, an engineering parameter, denotes the basic slot size.

To create a schedule, the GO requires data load information of each client. This information is sent upon group formation. Once the schedule is created, the GO starts a `schedule` thread which informs the GO about sender and receiver of the scheduled next transmission. To notify a client to send (or receive) data, the GO sends a control message `RequestSend` (or `RequestReceive`) to this client. Once receiving a confirmation message `ConfirmSend` (or `ConfirmReceive`), data can be transmitted. Upon receiving or sending the first data byte, the `schedule` thread starts a timer and will notify the GO to stop the ongoing transmission when the current slot is being used up. The client will be also informed by the GO with a `RequestPause` message. When a `ConfirmPause` message is received, the GO starts the next transmission. To avoid empty channel utilization, the GO will immediately inform the sender or receiver of the next transmission to start if the data of some node finishes before its slot ends. Fig. 2 provides an illustration of the work flow of PBS transmission scheduling during two consecutive sub-slots. In the first sub-slot, a client uploads its data to the GO, and in the second sub-slot, the GO forwards the received data to another client.

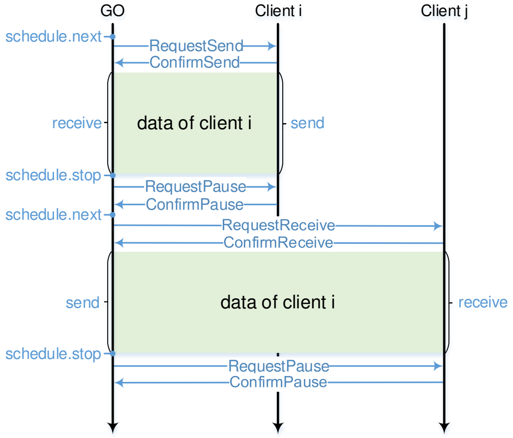


Fig. 2: Work flow of PBS transmission scheduling.

Fig. 3: The testbed used in our experiments.

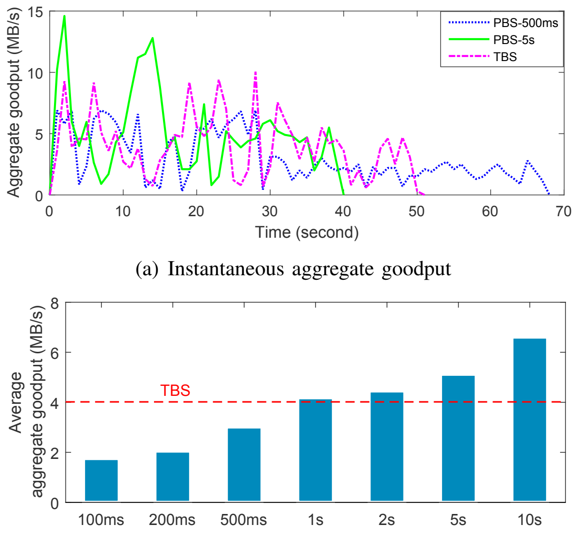
IV. Experimentation

Through various experiments, we investigate the goodput and fairness of TBS and PBS in local data dissemination, and seek the cases where PBS can possibly outperform TBS.

A. Experimental Setup

We setup a WiFi Direct testbed comprising three Nexus 6P phones running Android 6.0.1, as shown in Fig. 3. All the phones support WiFi Direct and have installed the MSN application. Group formation follows the procedures of the standard mode. The IP address of the GO is constantly 192.168.49.1 regardless of which phone is acting the GO, while that of the clients are allocated in range 192.168.49.x by the GO using the Dynamic Host Configuration Protocol (DHCP).

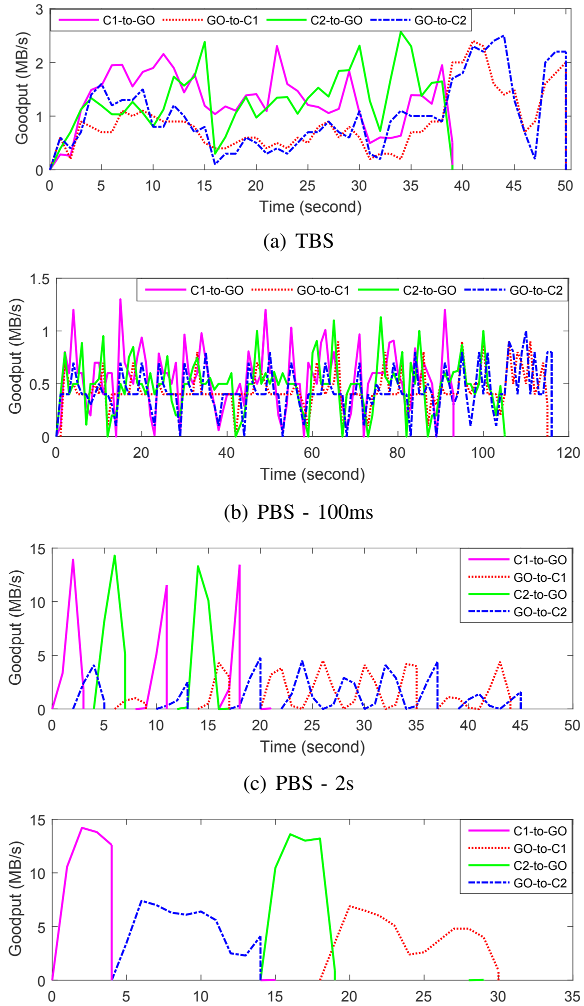
Theoretically, the devices exchange a GO Intent value in order to decide their roles in the group to be formed, and the one declaring the highest value becomes the GO [7]. In the experiments, however, we are able to decide which device to be the GO. Take two of the devices (named 'msn2' and 'msn3') we have as an example. When WiFi signal is turned on, 'msn2' and 'msn3' will appear on the list of available WiFi Direct peer devices shown on the other's device. Clicking 'msn3' on 'msn2's device will send an invitation to 'msn3'. Once the invitation is accepted, 'msn3' will automatically become the GO while 'msn2' will be the client. For convenience, we fix a device as the GO, and the other two devices as clients throughout all the experiments. The IP addresses of



(a) Instantaneous aggregate goodput

(b) Average aggregate goodput

Fig. 4: Aggregate goodput under TBS and PBS with different slot sizes.



(a) TBS

(b) PBS - 100ms

(c) PBS - 2s

(d) PBS - 10s

Fig. 5: Goodput under TBS and PBS with different slot sizes.

the clients are 192.168.49.253 and 192.168.49.85, which are kept unchanged during the experiments.

In the experiments, we measure the data goodput of the WiFi Direct group under various conditions. Specifically, we conduct three sets of experiments to study the impact of basic slot size τ, data load, and mobility on the performance of TBS and PBS. Each set of experiments are performed either after 5pm on the same day or on weekend when the interference is relatively low. As all data transmissions involve the GO, data goodput is measured at the GO. One data entry is recorded per second, which contains time stamp, the amount of data (in bytes) sent during last second, and the IDs of its sender and receiver. In the following, we present the experimental results. Note that the results in different sets of experiments may not be comparable with each other, as they are not obtained at the same time.

B. Slot Size

This set of experiments involve GO and two clients named C1 and C2. They are put close to each other on a desk. The data loads of C1 and C2 are both $50MB$. GO forwards data for each client to the other client. Fig. 4 shows the aggregate goodput under TBS and PBS with slot size τ varying from $100ms$ to $10s$. From the figure, we can see that 1) though the instantaneous goodput fluctuates over time, PBS with larger slot size results in shorter data dissemination time and consequently larger average aggregate goodput; and 2) PBS with large slot sizes (i.e. $1s, 2s, 5s$, and $10s$) tends to outperform TBS in terms of average aggregate goodput.

These observations can be explained by the slow-start behavior of TCP congestion control. Fig. 5 shows the goodput of each individual node (or TCP connection). Take Fig. 5(d) for example, which shows the goodput under PBS with $10s$ slot size. The transmission of each connection starts with low rate and reaches top rate after roughly $2s$. The rate can stay top during each slot due to no congestion and little channel contention within the group (the receiver may content with the sender to send ACK messages.). For PBS with $2s$ slot size (Fig. 5(c)), the rate can reach as high as PBS with $10s$ slot size can, however, it is unable to keep since the slot size (i.e. $2s$) does not allow. The frequent oscillation of send rates of all connections results in low average goodput. This impact is clearer when the slot size is small. From Fig. 5(b), it can be seen that the rate is not able to reach $1.5MB/s$ when the slot size is $100ms$. Due to channel contention, the goodput of each connection under TBS is low, as shown in Fig. 5(a). However, the aggregate goodput is higher than PBS with small basic slot sizes (i.e. $100ms$, $200ms$, and $500ms$).

Fig. 6 compares Jain's fairness index of TBS and PBS with different slot sizes. As can be seen, perfect fairness can be achieved by all if time allows all the data to be disseminated. In reality, however, the contact duration among a group of nodes can be so limited that not all the data can be disseminated. A study shows that the average duration of pedestrians with a mean speed of $1.3m/s$ is below $10s$ [11], which means at most

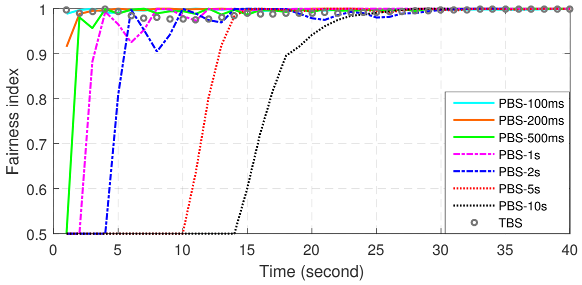


Fig. 6: Jain's fairness index of TBS and PBS with different slot sizes.

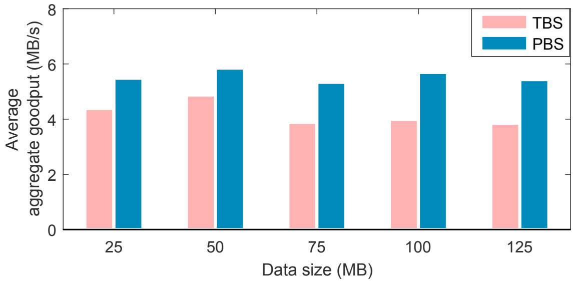


Fig. 7: Goodput under different data load of C2.

$50MB$ data on average can be transmitted during a contact, assuming the data rate is $5MB/s$ and no loss. Therefore, instantaneous fairness performance is more meaningful than the convergent fairness in the context of MSNs. We can observe from Fig. 6 that PBS with larger slot size, though providing higher aggregate goodput, tend to result in worse fairness in the first $20s$ wherein the contact is likely to end according to [11]. Clearly, there must be a trade-off between goodput and fairness in choosing a proper slot size for PBS. From Fig. 4 and Fig. 6, it can be seen that PBS with $2s$ slot size slightly outperforms TBS in both aggregate goodput and fairness. In the following experiments, we set the basic slot size of PBS to $2s$ to see if it still outperforms TBS.

C. Data Load

Fig. 7 illustrates the average aggregate goodput when the data load of C2 is set to $\{25, 50, 75, 100, 125\}MB$ and that of C1 is fixed to $50MB$. As can be seen, the goodput of TBS and that of PBS are not affected much. Given that the basic slot size is $2s$, both C1 and C2 are allocated by PBS a slot size that allows it to obtain a high data rate during every slot (cf. Section III-B), which explains higher goodput of PBS than TBS.

Fig. 8 compares the fairness indices of TBS and PBS under different loads of C2. On the one hand, it is shown that there are stronger oscillations in the fairness indices of PBS with larger data load of C2 in $(0, 30)s$. Such oscillation is a result of slot allocation by PBS. Given the basic slot size is $2s$, the whole slot sizes for disseminating the data of C2 with loads of $\{25, 50, 75, 100, 125\}MB$ are $\{2, 4, 6, 8, 10\}s$, respectively. Therefore, larger data load allows C2 to occupy the channel longer and consequently disseminate more data, causing stronger oscillation. On the other hand, since TBS provides throughput fairness (all sending nodes receive equal throughput), the data dissemination rates of C1 and C2 are approximately equal at the start, which causes more unfairness (by our definition) with more asymmetric data loads.

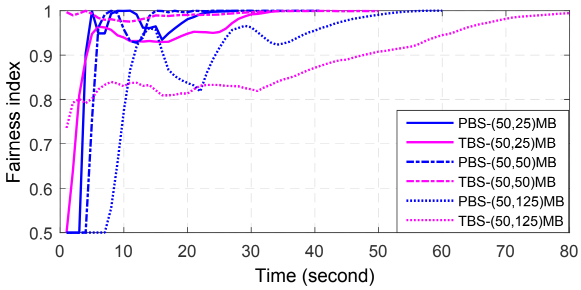
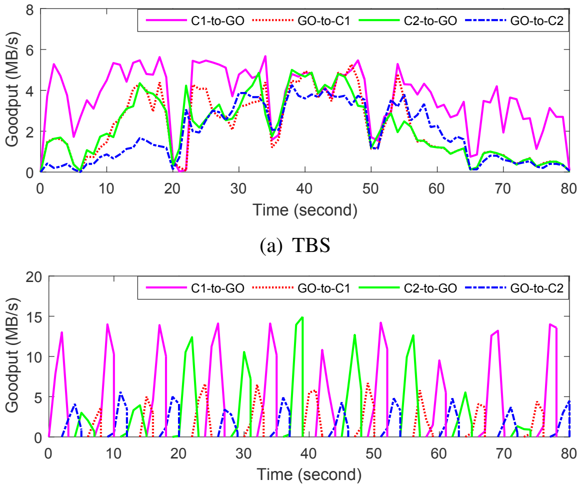


Fig. 8: Jain's fairness index under different data load of C2.

D. Mobility

In this set of experiments, we study the impact of mobility on the performance of TBS and PBS. Nodes have different distances to the GO when they move. Far node tends to have low signal strength and therefore low transmission rate to the GO. In traditional WiFi networks, it is known that performance (or rate) anomaly happens when nodes send data to the same AP at different transmission rates [12]. It means that the low-rate node occupies much longer airtime than the high-rate node due to the throughput fairness nature of DCF, and thereby the throughput of high-rate node is dramatically compromised. To figure out if such phenomenon also exists in WiFi Direct networks, which may possibly affect the performance of TBS and PBS, we perform several tests where the GO and C1 move side by side from one end to the other end of a corridor of $50m$ at a low speed of roughly $0.6m/s$, while C2 is fixed at the middle of the corridor. A low speed enables us to observe the goodput change in fine granularity. In addition, the data loads of C1 and C2 are both set to $500MB$, so that we are able to observe the goodput change of every connection throughout each test. Fig. 9 shows the typical instantaneous goodput of individual connections during one test. In general, performance anomaly is not seen in both TBS and PBS, as C1, which is always close to the GO, obtains much higher goodput than C2 when it is far from the GO.

Knowing that there is no much evidence of performance anomaly, we conduct experiments with a more realistic setting where the GO and one client move together along the corridor at a walking speed of roughly $1.6m/s$, while the other client is fixed at the middle of the corridor. We consider the following three cases 1) data loads of (C1, C2) are $(50, 50)MB$ and C1 moves with the GO; 2) data loads are $(50, 100)MB$ and C1 moves with the GO; and 3) data loads are $(100, 50)MB$ and C2 moves with the GO. Fig. 10 shows that PBS in general provides higher goodput than TBS. A comparison on the fairness indices of TBS and PBS in the three cases is provided in Fig. 11. It shows that the fairness indies of TBS and PBS are relatively high when C2 who has heavy data load moves with the GO, whilst they are low (especially TBS) when C1 which has light data load moves together with the GO. It indicates that high data load combined with low goodput results in low fairness regarding dissemination rate.



(a) TBS

(b) PBS - 100ms

Fig. 9: Instantaneous goodput under TBS and PBS when the GO and C1 move.

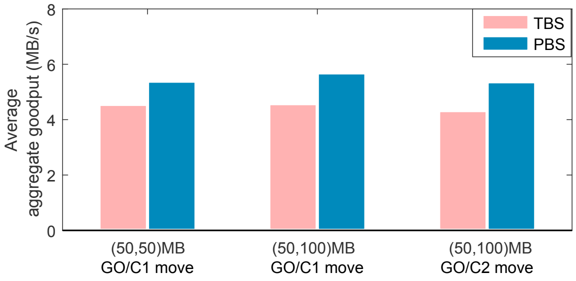


Fig. 10: Aggregate goodput vs. mobility

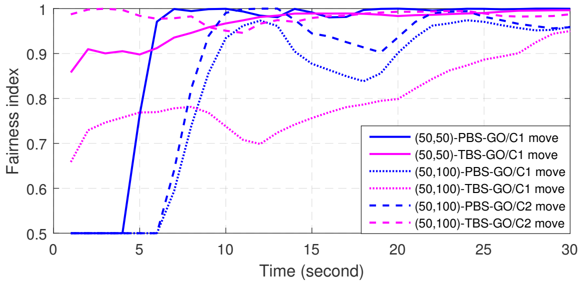


Fig. 11: Jain's fairness index vs. mobility

E. Summary

The experimental results show that WiFi Direct is able to provide an average goodput of more than $4MB/s$. Notably, PBS with large basic slot sizes (i.e. $1s, 2s, 5s$, and $10s$) can achieve higher goodput than TBS, due to its capability of avoiding channel contention on the application layer. The results also suggest that the fairness index of TBS is considerably sensitive to data loads and mobility of nodes. Specifically, its fairness index is lower than PBS with $2s$ basic slot size (except in the first few seconds) when the data loads of nodes are asymmetric (Fig. 8), and it becomes worse if the node with relatively large data load receives low goodput (Fig. 11). In fact, the low fairness index of PBS in the first few seconds and large oscillation is a result of the specific implementation of transmission scheduling (i.e. distributing the sub-slots of one node in a row). It can be improved by distributing the nodes' sub-slots dispersedly.

Leaving transmission scheduling to DCF that is at the MAC layer, TBS is only able to provide best-effort service. On the contrary, PBS provides a simple framework for network optimization to developers through its functions of transmission scheduling and slot allocation. They can implement the scheduling algorithm and slot allocation scheme that optimize their own objectives (e.g. content prioritization) on data dissemination.

V. Conclusion

In this paper, we experimentally evaluated the performance of WiFi Direct for local data dissemination in MSNs. We developed an MSN application that enables data dissemination among a group of WiFi Direct nodes with two types of transmission strategies, i.e. TBS and PBS. The results of the experiments indicate that PBS with large basic slot size can achieve higher aggregate goodput than TBS. In addition, the fairness index of TBS is more easily affected by data loads and mobility than that of PBS. In our future work, we would like to improve the fairness index of PBS by distributing the nodes' sub-slots dispersedly and investigate UDP goodput and fairness with more nodes.

Acknowledgment

The work was partly supported by the EU FP7 CLIMBER project (www.fp7-climber.eu). The authors would like to thank Kristjan Reinloo for improving the implementation.

References

[1] W.-F. Alliance, "Wi-Fi CERTIFIED Wi-Fi Direct® : Personal, portable Wi-Fi® technology," *White Paper*, 2014.

[2] C. Funai, C. Tapparello, and W. B. Heinzelman, "Supporting multi-hop device-to-device networks through wifi direct multi-group networking," arXiv:1601.00028, 2016.

[3] C. Casetti, C. F. Chiasserini, L. C. Pelle, C. Del Valle, Y. Duan, and P. Giaccone, "Content-centric routing in wi-fi direct multi-group networks," in *IEEE WoWMoM*, 2015, pp. 1–9.

[4] A. A. Shahin and M. Younis, "Efficient multi-group formation and communication protocol for wi-fi direct," in *IEEE LCN*, 2015.

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

[6] G. Tan and J. V. Guttag, "Time-based fairness improves performance in multi-rate wlans." in *USENIX ATC*, 2004, pp. 269–282.

[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, 2013.

[8] R. Chandra, S. Karanth, T. Moscibroda, V. Navda, J. Padhye, R. Ramjee, and L. Ravindranath, "Dircast: A practical and efficient wi-fi multicast system," in *IEEE ICNP*, 2009, pp. 161–170.

[9] "Point coordination function," https://en.wikipedia.org/wiki/Point_coordination_function.

[10] R. Jain, A. Durresi, and G. Babic, "Throughput fairness index: An explanation," Tech. rep., The Ohio State University, 1999.

[11] Ó. Helgason, S. T. Kouyoumdjieva, and G. Karlsson, "Opportunistic communication and human mobility," *IEEE Transactions on Mobile Computing*, vol. 13, no. 7, pp. 1597–1610, 2014.

[12] P. Bahl, R. Chandra, P. P. Lee, V. Misra, J. Padhye, D. Rubenstein, and Y. Yu, "Opportunistic use of client repeaters to improve performance of wlans," *IEEE/ACM Transactions on Networking (TON)*, vol. 17, no. 4, pp. 1160–1171, 2009.