Application Level Quality Measurement of Heterogeneous Device-to-Device Links

Md Tausif Al Hossain[†], Mohammad Arif Hossain[‡], and Murat Yuksel[‡,†]
Computer Science[†], Electrical & Computer Engineering[‡], University of Central Florida, Orlando, FL, USA
tausif.hossain@Knights.ucf.edu, arif.hossain@knights.ucf.edu, murat.yuksel@ucf.edu

***Abstract*—Device-to-device (D2D) links enable direct communication between mobile devices without using the cellular network or the Internet. This type of communication can be helpful in situations when there is a partial or complete failure of the network infrastructure. However, effective use of D2D links for a multi-hop D2D communication system requires quick and practical quantification of their quality from user space of devices. We developed an Android application to search and measure the quality of available D2D links nearby. This application can search for heterogeneous (i.e., Bluetooth and WiFi Direct) D2D links simultaneously and measure the link quality from the user space. We run indoor and outdoor experiments to examine how line-of-sight or non-line-of-sight D2D links perform.**

***Index Terms*—D2D, Bluetooth, WiFi Direct, Android, Link Quality, Heterogeneous, Infrastructure-less**

I. Introduction

Natural disasters are an inseparable part of our life and the environment. The 20th and 21st century have had many devastating disasters that wiped out thousands of invaluable lives. More than 35,000 natural disasters took place since the beginning of the 20th century causing the death of more than 8 million people [1]. Many of them could be saved if there were adequate assistance during and after the disasters.

In order to save lives during a disaster, it is imperative to establish effective communication to let the first responders, e.g., Emergency Medical Services (EMS), police, fire rescue, and 911 know about the location of the victims. However, the help of the first responders might not be sufficient as there could be lots of people seeking help in numerous places at the same time. Moreover, the disaster can damage the communication infrastructure. Social media like - Twitter, Facebook and Instagram have become alternative platforms in such scenarios. During the tsunami at Japan in 2011 and hurricanes Harvey and Irma, people mainly used Twitter, to keep the respective organizations and officials posted about their situation and to seek help from others [2].

During disasters, major failures may happen in cellular communication and the Internet infrastructure. Device-to-device (D2D) communication is an alternative in such situations to establish communication. This technology has flourished a lot for messaging and data sharing among devices with assistance from cellular and heterogeneous networks [3]–[5]. However, D2D communication without infrastructure support is a recent topic for disaster management.

End-to-end connectivity using D2D links will have to adapt for infrastructure-less multi-hop operation in case of disaster management. Effective multi-hop D2D communications require (i) seamless operation on devices from different vendors and (ii) swift detection of good quality links. To address these issues, we develop an user space Android application and implement link quality measurement functions within the application. We implement the application with searching and quality measuring functions for heterogeneous D2D links [6]. We use both WiFi Direct and Bluetooth modules to enhance the possibility of getting connected in infrastructure-less conditions. Major contributions include:

- Simultaneous discovery of WiFi Direct and Bluetooth peers in Android.
- User-space methods to measure received signal strength indicator (RSSI), throughput and effective round-trip time (RTT) for single-hop D2D links.
- Extensive experiments in indoor, outdoor, line-of-sight (LoS) and non-line-of-sight (NLoS) cases using devices from various vendors.
- Analysis of correlation among different D2D link quality metrics and analysis of convergence to correct metric value.

The key insights we gained from our D2D link quality measurement experiments are:

- There is notable difference among Bluetooth and WiFi Direct in peer search process, how D2D connections are formed, and how data is transferred.
- RSSI is a more reliable metric than RTT and throughput in different environments. Yet, RSSI is only available on Bluetooth API in Android.
- Bluetooth RSSI is sufficient to determine the quality of the WiFi Direct link (between the same devices) in terms of (i) TCP throughput in all environments, (ii) UDP throughput for indoor NLoS or outdoor LoS cases, and (iii) RTT in outdoor LoS case.
- Sub-second quantification of the D2D link's quality is feasible with hybrid methods utilizing RSSI and UDP-based throughput measurements.

The rest of this paper is organized as follows. Section II

Tausif Al Hossain is now affiliated with JABIL Healthcare. He was with UCF during the course of this work.

Arif Hossain is now affiliated with New Jersey Institute of Technology. He was with UCF during the course of this work.

This work was supported in part by NIST PSIAP grant 70NANB17H188.

978-1-7281-8154-7/20/$31.00 ©2020 IEEE

covers the related works. Section III describes the peer discovery process. Section IV presents the metrics and methods used for D2D link quality measurements. Section V shows the experiments and results of D2D link quality measurements. Finally, Section VI concludes our work with future directions.

II. Related Works

Since D2D communication has become an accessible technology of the future generation wireless networks, many researchers have proposed various schemes using D2D techniques. Network-assisted D2D communication is one of the promising technologies to maintain communication during disaster management. Ali *et al.* proposed a D2D network architecture for public safety [7], which used LTE network for supporting the D2D communication protocol. Although the architecture is useful for disaster relief and public safety, the D2D communication mechanism would not sustain without the assistance of the LTE network. Another D2D system for disaster communication was presented using a multi-hop D2D relay and introducing the 4G and 5G cellular network as the support network [8]. Several other works have also been proposed for disaster management based on network-assisted D2D communication [9]–[12]. Those works are mainly based on cellular, cognitive, and Software-Defined Networks requiring infrastructure support.

Recently, there has been more interest for D2D systems not requiring infrastructure support. Moghaddam *et al.* proposed a disaster-response network architecture based on D2D communication without the support of cellular infrastructure illustrating the D2D communication method using the relay and cooperative beamforming [13]. The authors proposed a multi-hop cooperative relay network based on a cluster-based hierarchical topology. Although the work described an infrastructure-less scenario for D2D communication, it did not utilize heterogeneous (both WiFi Direct and Bluetooth) wireless interfaces for establishing a D2D network. Other studies looked at the feasibility of opportunistic multi-hop phone-to-phone communications [14] and group formation and power consumption aspects of WiFi Direct for D2D communications [15].

With the motivation to reduce dependence on the infrastructure, sharing of spectrum resources [16], [17] as well as content or services among devices received notable attention. Sharing data [18], [19] and connectivity services among user applications running on different devices can help notably for disaster networking. Rahman *et al.* proposed a service sharing application using heterogeneous D2D links without infrastructure support [20]. Although these studies proposed a multi-hop architecture, they do not include peer discovery and measuring of D2D link quality measurement so as to help making multi-hop routing decisions.

Measuring the quality of individual links is a critical aspect for multi-hop D2D systems. Mtibaa *et al.* experimented on D2D communication using Internet-of-Things (IoT) devices in indoor and outdoor environments [21]. Although the authors used both Bluetooth and WiFi Direct, they did not implement simultaneous peer discovery. Further, the time to measure the link quality was not investigated. In our work, we develop an Android application using both Bluetooth and WiFi Direct links. We implement simultaneous peer discovery for heterogeneous links and develop techniques for measuring the quality of those links with a focus on link quality detection time. Although we work with single hop links, routing and multi-hop communication can be developed on top of this application.

III. Simultaneous Peer Discovery

We implement a proactive approach to discover heterogeneous peers simultaneously. For Bluetooth links, we implement periodic discovery which divides the running time of the application into equal time slots. Discovery runs and stops on alternating time slots. We set the time slot length to 10 seconds in our experiments as a simplifying assumption. Our experiments reveal that most of the devices are not visible unless the Bluetooth settings page is open on the phone. To solve this, we use 'discoverable' option which allows the device to be visible up to 3600 seconds.

For WiFi Direct discovery, we observe that two devices discover each other only when they run the discovery simultaneously. If we use periodic approach, it will stop discovery in alternating time slots and make the device invisible. As a result, we implement the continuous WiFi Direct discovery to make devices always visible.

Bluetooth and WiFi Direct have dedicated threads for peer discovery. The overall process is shown in Fig. 1. Home activity controls the user interface of the application and maintains a peer list. It initiates the peer discovery controller which starts a broadcast receiver, WiFi Direct discovery, and Bluetooth discovery modules. WiFi Direct and Bluetooth discovery modules notify the broadcast receiver about any change in the peer list. The broadcast receiver updates a temporary peer list maintained by peer discovery controller. Bluetooth discovery controller controls the Bluetooth discovery and sends the updated peer list periodically to the home activity.

IV. D2D Link Quality Measurements

A. Why Measure D2D Link Quality Fast?

D2D link quality measurement is crucial for two reasons: First, multiple devices form a group with a master and multiple slaves when they are peered together. Two slave devices communicate through the master forming a low quality two-hop link. An illustration of this scenario is shown in Fig. 2.

Second, the peer discovery process can generate many D2D links. It is critical to filter out bad links to choose the best routing path.

B. Metrics and Methods

We measure effective RTT and throughput of Bluetooth and WiFi Direct links in indoor and outdoor environments. For Bluetooth links, we measure the RSSI as well. RSSI provides a very precise measurement of the link quality; however, since it is based on physical layer measurement only, it does not account for other delays and dynamic bottlenecks involving the operating systems of the devices. Hence, we

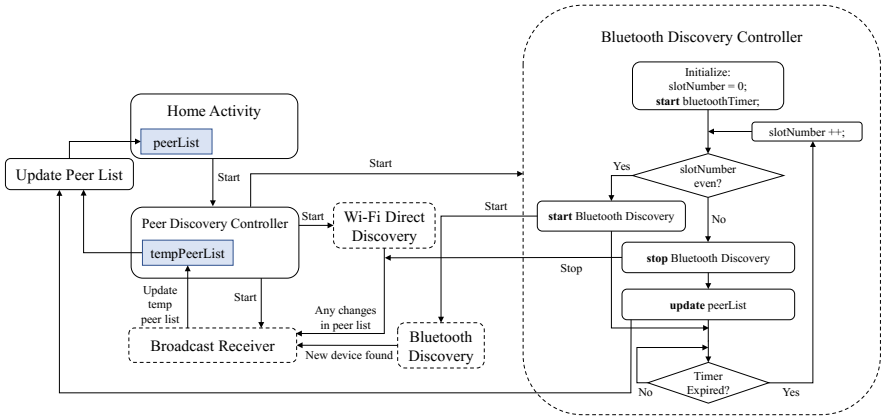


Fig. 1. Peer discovery (dashed boxes denote the threads running in the background)

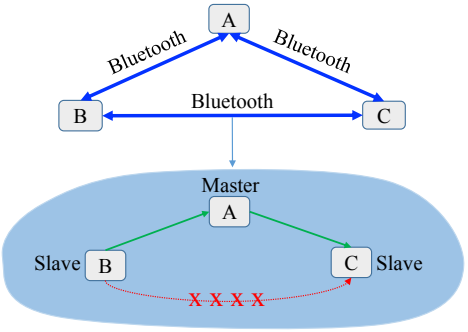


Fig. 2. Intra-group communication of D2D links: Despite appearing as a single link to user-space application, intra-group communication for a D2D link may involve two hops if it is between two slave nodes.

(a) Indoor LoS (b) Outdoor LoS (c) Indoor NLoS (d) NLoS Setup

Fig. 3. Experimental environments

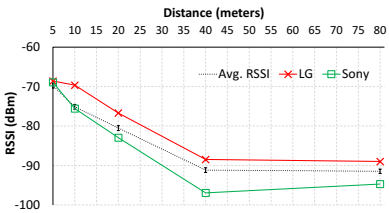


Fig. 4. Bluetooth RSSI at outdoor

devise techniques to measure the effective throughput and RTT between the application processes on the peer devices. For each experiment, we first peer two devices from different vendors, and then, perform the measurements. Throughout the paper, the measuring device is referred as the *sender* and the device being measured is referred as the *receiver*.

1) Measuring RSSI: It is possible to measure RSSI of a Bluetooth link but not possible for a WiFi Direct link from available APIs. Our investigation reveals that RSSI is hardcoded in the WiFi Direct library [22] and is not available to user-space applications. Android Bluetooth API comes with the function to search for Bluetooth devices nearby. When a device is found the search function returns its name, MAC address and RSSI. So, we measure RSSI during the peer discovery for Bluetooth links.

2) Measuring RTT: To measure RTT for Bluetooth, we use RFCOMM channel provided in Android API. We use TCP-like reliable transfers to measure RTT as Android only supports TCP-like sockets for Bluetooth [23], [24]. To measure RTT, the sender sends a 500B packet and the receiver echoes the packet back. The sender calculates RTT from the time difference between the echo and the packet. Since the Bluetooth sockets are reliable, we do not consider the possibility that the packet or its echo are lost. However, the TCP-like socket may be retransmitting the packet due to loss and may cause the measured RTT to be more than the actual RTT. Since the D2D link effectively includes these retransmissions, we consider them as part of the RTT.

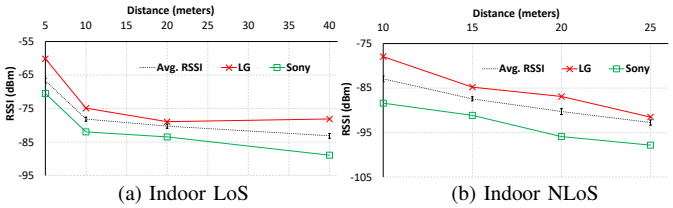
For WiFi Direct, we use unreliable sockets to measure RTT. The sender sends a packet and waits for a time T_{RTT} to get a response. If received successfully, the receiver echoes the packet back. If the sender receives the echo before T_{RTT} expires, it considers it valid and calculates the RTT from the time difference between the echo and the packet. The use of T_{RTT} truncates outliers from the samples and makes the RTT more reliable. We set T_{RTT} = 1 s in our experiments.

3) Measuring Throughput: We only measure throughput for WiFi Direct links as RSSI is already available for Bluetooth links. The golden standard for measuring effective throughput is TCP. So, we measure throughput by sending a 10 MB file over a TCP connection. The receiver records how long it takes to receive the file and calculates throughput by dividing the file size by the time to receive the file. In our experiments, it took more than 10s for WiFi Direct links to complete the TCP-based file transfers. Hence, faster throughput measurement is needed for our goal of quantifying the link quality quickly, preferably in the order of milliseconds.

We use UDP traffic to get an understanding of the effective throughput. We send two UDP packets of *different* sizes and calculate the RTT for each by waiting for their echos from the receiver side. Let the packet sizes be P_1 and P_2 and RTT values be RTT_1 and RTT_2, then we calculate effective throughput R by solving the following equations

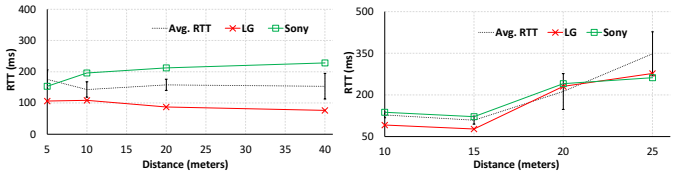
$$RTT_1 = 2(d + P_1/R), \quad RTT_2 = 2(d + P_2/R), \qquad (1)$$

where d is the one-way propagation delay of the D2D link. Since d and R are the only unknowns, we solve the equations for these unknowns and calculate R after every successful transmissions of two subsequent packets. We keep sending a sequence of such UDP packets and collect more throughput samples until their average converge, which will be detailed in Sec. V-D. This approach considers other internal node delays as part of the effective *per-packet throughput* and notably will yield an imprecise measure of the actual throughput. Since



(a) Indoor LoS (b) Indoor NLoS

Fig. 5. Bluetooth RSSI measurements for indoor LoS and indoor NLoS



(a) Indoor LoS (b) Indoor NLoS

Fig. 6. Bluetooth RTT measurements for indoor LoS and indoor NLoS

TABLE I
DEVICE CONFIGURATIONS

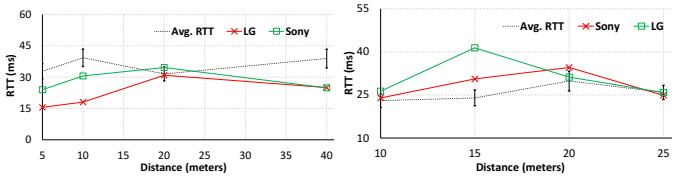
Vendor	Model	Processor	RAM
LG	Nexus 5	2.26 GHz Quad core	2GB
Motorola	Moto E4	1.4 GHz Quad core	2GB
Sony	Xperia L1	1.45 GHz Quad core	2GB
Nokia	Nokia 2	1.3 GHz Quad core	1GB

our goal is a quick and rough quantification of the D2D link quality, we focus on, instead of measuring the actual throughput, getting a UDP-based throughput measurement which will be in high correlation with the TCP-based one.

V. RESULTS

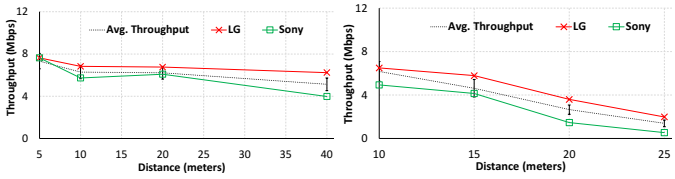
A. Environment and Setup

We perform experiments in both indoor (line-of-sight (LoS) and non-line-of-sight (NLoS)) and outdoor (LoS only) environments using devices from different vendors. We use approximately a 60m long and 3m wide corridor for the indoor LoS experiments and a corner on the corridor for the NLoS measurements. The indoor LoS and NLoS environment are shown in Fig. 3(a) and Fig. 3(c), respectively. In case



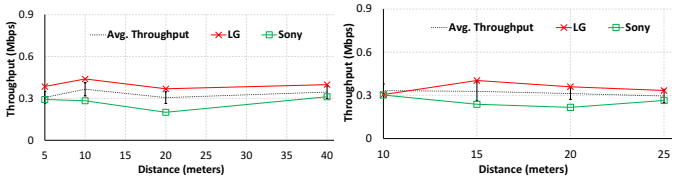
(a) Indoor LoS (b) Indoor NLoS

Fig. 7. WiFi Direct RTT measurements for indoor LoS and indoor NLoS



(a) Indoor LoS (b) Indoor NLoS

Fig. 8. WiFi Direct: TCP-based throughput measurements



(a) Indoor LoS (b) Indoor NLoS

Fig. 9. WiFi Direct: UDP-based throughput measurements

of NLoS experiments, we divide the link distance c equally in both sides of the corner, i.e., a=b=$c/2$ in Fig. 3(d). We use a field for the outdoor experiments as shown in Fig. 3(b) and measure up to 80 m. We use four Android devices from different vendors with various configurations as shown in Table I. We run each experiment between two devices and consider one as the sender and another as receiver. We perform the experiments multiple times (50+ times for RSSI, RTT and UDP-based throughput measurements; and 15+ times for TCP-based throughput measurements) and plot the average values as a function of distance with 90% confidence intervals.

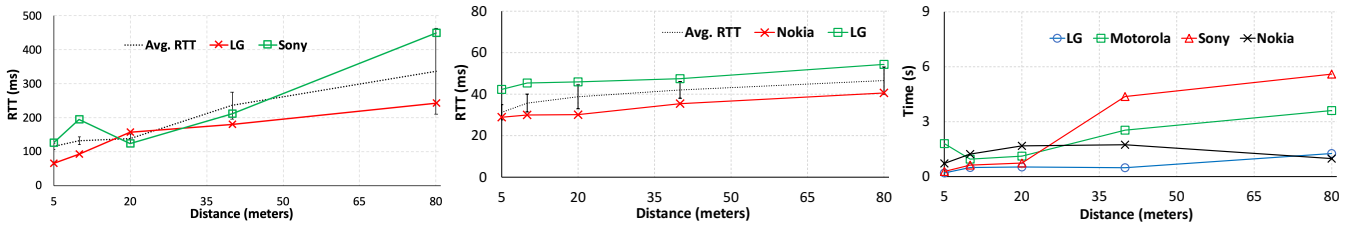
B. Bluetooth Experiments

1) RSSI: The Bluetooth RSSI results for each device as average value, best value, and worst value for indoor LoS and NLoS cases are shown in Fig. 5 while Fig. 4 shows the outdoor case. As expected, the outputs indicate that RSSI decreases with increasing distance. It is visible from the results that RSSI is stronger in the indoor LoS environment than outdoor LoS environment due to multipath propagation. Besides, RSSI in indoor NLoS environment gets weaker than both the indoor and outdoor LoS environments. We also find that LG and Sony as the best and worst performing devices, respectively.

2) RTT: The average RTT found from all the vendors along with the best and worst vendors for the indoor LoS and NLoS are shown in Fig. 6(a) and (b), respectively; and the outdoor LoS case is shown in Fig. 10(a). These outcomes are in alignment with the RSSI results and show that RTT is mostly increasing with distance in all the environments and the average RTT measurements are in milliseconds. It also evident again that LG and Sony are also the best and worst performers, respectively.

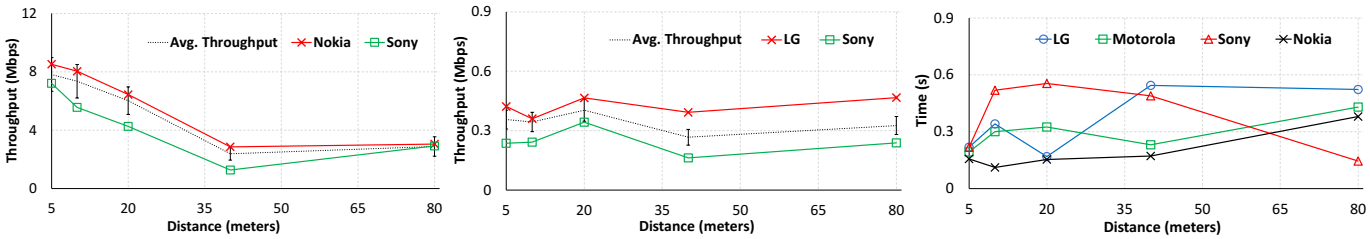
C. WiFi Direct Experiments

1) RTT: We use 500B packets to measure RTT of WiFi Direct links. Fig. 7 shows the average, best and worst RTT values in indoor LoS and NLoS cases, where we observe that RTT is not always increasing with the increase in the distances. This likely caused by the multi-path fading from the reflections from walls. For the outdoor, we do not observe such variation and RTT mostly increase with distance as shown in Fig. 10(b). We observe that LG and Sony work as the best and worst vendors, respectively for the indoor LoS setup while Sony outperforms other vendors and LG performs as the worst at the indoor NLoS case. Nokia and LG operates as the best and worst vendors, respectively for the outdoor environment.



(a) Bluetooth RTT (b) WiFi Direct RTT (c) Bluetooth RTT convergence

Fig. 10. Bluetooth and WiFi Direct RTT and Bluetooth RTT convergence results in outdoor LoS environment



(a) TCP Throughput (b) UDP Throughput (c) UDP Throughput Convergence

Fig. 11. WiFi Direct: TCP throughput, UDP throughput, and UDP throughput convergence outputs for outdoor LoS

2) Throughput: As expected, measurement results show that TCP-based throughput decreases with increasing distance. Figure 8 shows average, best and worst TCP throughput outputs for the indoor LoS and NLoS while Fig. 11(a) shows the TCP throughput results of the outdoor case. The figures reveal that the highest average TCP throughput from all the vendors is 7.82 Mbps at 5 m distance in the outdoor case whereas the lowest average TCP throughput from all the vendors is 1.38 Mbps at 25 m distance in the indoor NLoS case. We also find LG and Sony as the best and the worst vendors, respectively in the indoor experiments. But, we find Nokia and Sony as the best and worst vendors, respectively for the outdoor case.

Fig. 9 and Fig. 11(b) illustrate that the highest average UDP-based per-packet throughput from all the vendors is 0.40 Mbps at 20 m distance in outdoor environment whereas the lowest average throughput is 0.27 Mbps at 40 m in outdoor environment. We observe that LG and Sony are the best and the worst vendors, respectively, in all cases. As we expected, the UDP-based throughput measurements are significantly lower than the TCP-based ones. This is mainly due to our approach measuring the per-packet throughput instead of a window-based approach which would utilize the full link throughput capacity via pipelining.

D. Convergence Analysis

It is crucial to reduce the time it takes to quantify the D2D links' quality so that routing decisions can be made quickly. We analyze the convergence time of the quality metrics in our measurements as this will show how long our measurements will have to continue before stably determining the quality of the link at hand. We estimate the metric value after the k^{th} measurement using Exponential Weighted Moving Average as:

$$M_{E_k} = M_{E_{k-1}}(1-\alpha) + M_k\alpha \quad (2)$$

where M_k is the measurement on the k^{th} iteration, and M_{E_k} and $M_{E_{k-1}}$ are the estimated metric values for k^{th} and $k-1^{\text{th}}$ iterations, respectively. The weight value α ranges between 0.01 and 0.99 with a step size of 0.01. For each α, we find the first iteration i where the estimate M_{E_i} is within an error margin of $\pm 5\%$ of average metric value M_{avg} observed at the end of our measurements. We thus find multiple i values corresponding to each α value and set the minimum i as the number of iterations to reach convergence.

We show the convergence analysis of Bluetooth RTT for the outdoor case in Fig. 10(c). Convergence time ranges between 1 to 6 s. Since Bluetooth packets transfers are via TCP-like sockets, it takes longer time to converge to a solid average RTT because of the retransmissions being counted as part of the RTT measurements. We show the convergence analysis for UDP-based throughput measurements in Fig. 11(c). To generate varying packet sizes so that (1) can be solved, we use random sized packets ranging from 50 B to 500 B. The convergence time is significantly reduced to sub-seconds for the UDP-based throughput measurement in comparison to Bluetooth RTT measurements. Further, the convergence time does not increase with larger distance. These are good traits of the UDP-based approach showing a promising prospect for quick link quality quantification. However, since this UDP-based per-packet throughput measurement approach is imprecise, we need to verify if it correlates with the TCP-based measurements, which is next.

E. Correlation Against Bluetooth RSSI

It is important to measure as few metrics as possible to quickly estimate the D2D link quality and make routing decisions. To this end, we perform correlation analysis between Bluetooth RSSI and the other metrics. Since Bluetooth range is shorter, we only consider the WiFi Direct measurements up to the maximum Bluetooth range when correlating the two. The

TABLE II
CORRELATION AGAINST BLUETOOTH RSSI

Environment	Metric	Correlation
Indoor LoS	Bluetooth RTT	0.77
	WiFi-Direct RTT	-0.46
	WiFi-Direct TCP Throughput	0.93
	WiFi-Direct UDP Throughput	-0.46
Indoor NLoS	Bluetooth RTT	-0.87
	WiFi-Direct RTT	-0.68
	WiFi-Direct TCP Throughput	0.99
	WiFi-Direct UDP Throughput	0.96
Outdoor LoS	Bluetooth RTT	-0.88
	WiFi-Direct RTT	-0.96
	WiFi-Direct TCP Throughput	0.98
	WiFi-Direct UDP Throughput	0.59

reasons for using Bluetooth RSSI as the basis are that (1) it is measured at the physical layer and any software stack delays or disruptions are excluded, and (2) our experiments indicated that Bluetooth RSSI shows a steady decreasing pattern with increasing distance, which is in alignment with the intuition that link quality should deteriorate with distance.

Table II shows the tabulated correlation against Bluetooth RSSI. First, we observe that RTT is clearly affected by system issues, not by the link's channel quality. This indicates that delay-sensitive routing decisions will be hard to make given that RTT is not a reliable quality metric. On the positive side, we observe in all cases that WiFi TCP throughput is in strong correlation with Bluetooth RSSI, verifying our intuition that Bluetooth RSSI and TCP-based throughput measurements are reliable methods of measuring the link quality. More importantly, we can just measure Bluetooth RSSI in order to infer the WiFi Direct link quality, effectively eliminating the need to measure the WiFi Direct link quality as soon as two devices establish a Bluetooth link - simply by reading the RSSI from Bluetooh API (Sec. IV-B1). This addresses the WiFi Direct link quality measurement at short distances, i.e., < 25m. However, for longer ranges where a Bluetooth signal cannot be captured, UDP-based throughput measurement is needed. Our current method (Sec. IV-B3) shows glimmers of hope as it has strong correlation (0.96) against Bluetooth RSSI for indoor LoS case and notably positive correlation (0.59) in the outdoor case. It is based on inspecting two subsequent packets, and can be improved by adopting better techniques in future.

VI. CONCLUSION

We have developed an user space application to search and measure quality of single hop Bluetooth and WiFi Direct links. Our study addresses the implementation challenges present in Android. Our empirical analysis shows how quickly we can reach to a good estimation of a D2D link quality. Our study also discusses how different performance metrics correlate among themselves. Our application can provide a foundation for establishing a multi-hop D2D communication system for infrastructure-less situation. Future work will focus on investigating link quality of multi-hop links and how does grouping among devices affect the performance of D2D links. Developing dynamic routing over multi-hop D2D links is another major challenge to address in the future.

REFERENCES

[1] Karlsruhe Institute of Technology, "Natural disasters since 1900: Over 8 million deaths, $7 trillion," www.sciencedaily.com/releases/2016/04/160418092043.htm, 2016.

[2] M. Jahanian, Y. Xing, J. Chen, K. K. Ramakrishnan, H. Seferoglu, and M. Yuksel, "The evolving nature of disaster management in the internet and social media era," in *Proc. of IEEE LANMAN*, 2018.

[3] J. M. B. da Silva, G. Fodor, and T. F. Maciel, "Performance analysis of network-assisted two-hop D2D communications," in *2014 IEEE Globecom Workshops (GC Wkshps)*, Dec 2014, pp. 1050–1056.

[4] J. Liu, N. Kato, J. Ma, and N. Kadowaki, "Device-to-device communication in LTE-advanced networks: A survey," *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 1923–1940, Fourthquarter 2015.

[5] L. Wei, R. Q. Hu, Y. Qian, and G. Wu, "Enable device-to-device communications underlaying cellular networks: challenges and research aspects," *IEEE Comm. Magazine*, vol. 52, no. 6, pp. 90–96, June 2014.

[6] "D2D Link Quality Measurement Android Application," https://github.com/tausif-ah, 2016.

[7] K. Ali, H. X. Nguyen, P. Shah, Q. Vien, and N. Bhuvanasundaram, "Architecture for public safety network using D2D communication," in *2016 IEEE Wireless Communications and Networking Conference (WCNC) Workshops*, April 2016, pp. 206–211.

[8] K. Ali, H. X. Nguyen, P. Shah, Q. Vien, and E. Ever, "D2D multi-hop relaying services towards disaster communication system," in *Proc. of International Conference on Telecommunications*, May 2017, pp. 1–5.

[9] M. Usman, A. A. Gebremariam, U. Raza, and F. Granelli, "A software-defined device-to-device communication architecture for public safety applications in 5G networks," *IEEE Access*, vol. 3, pp. 1649–1654, 2015.

[10] H. Ahmad, M. Agiwal, N. Saxena, and A. Roy, "D2D-based survival on sharing for critical communications," *Wireless Networks*, vol. 24, no. 6, pp. 2283–2295, Aug 2018. [Online]. Available: https://doi.org/10.1007/s11276-017-1469-2

[11] K. Ali, H. X. Nguyen, Q. Vien, P. Shah, and Z. Chu, "Disaster management using D2D communication with power transfer and clustering techniques," *IEEE Access*, vol. 6, pp. 14 643–14 654, 2018.

[12] J. Z. Moghaddam, M. Usman, F. Granelli, and H. Farrokhi, "Cognitive radio and device-to-device communication: A cooperative approach for disaster response," in *Prof. of IEEE GLOBECOM*, Dec 2016, pp. 1–6.

[13] J. Z. Moghaddam, M. Usman, and F. Granelli, "A device-to-device communication-based disaster response network," *IEEE T. on Cognitive Communications & Networking*, vol. 4, no. 2, pp. 288–298, June 2018.

[14] S. Liu and A. D. Striegel, "Exploring the potential in practice for opportunistic networks amongst smart mobile devices," in *Proc. of ACM MOBICOM*, New York, NY, USA, 2013, pp. 315–326.

[15] 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.

[16] D. Wu, J. Wang, R. Q. Hu, Y. Cai, and L. Zhou, "Energy-efficient resource sharing for mobile device-to-device multimedia communications," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 5, pp. 2093–2103, Jun 2014.

[17] R. Zhang, X. Cheng, L. Yang, and B. Jiao, "Interference-aware graph based resource sharing for device-to-device communications underlaying cellular networks," in *Proc. of IEEE WCNC*, April 2013, pp. 140–145.

[18] X. Wang, Y. Zhang, V. C. M. Leung, N. Guizani, and T. Jiang, "D2D big data: Content deliveries over wireless device-to-device sharing in large-scale mobile networks," *IEEE Wireless Communications*, vol. 25, no. 1, pp. 32–38, February 2018.

[19] H. Wang, X. Wang, K. Li, J. Ren, X. Zhang, and T. Jiang, "A measurement study of device-to-device sharing in mobile social networks based on Spark," *Concurrency and Computation: Practice and Experience*, vol. 29, no. 16, p. e4021, 2017.

[20] M. Rahman, S. Mathew, M. Yuksel, and S. Sengupta, "A device-to-device service sharing middleware for heterogeneous wireless networks," in *Proc. of IEEE LANMAN*, June 2016, pp. 1–6.

[21] A. Mtibaa, A. Emam, S. Tariq, A. Essameldin, and K. A. Harras, "On practical device-to-device wireless communication: A measurement driven study," in *Proc. of International Wireless Communications and Mobile Computing Conference (IWCMC)*, June 2017, pp. 409–414.

[22] "Android WifiP2pPeer Class," https://android.googlesource.com/platform/packages/apps/Settings/+/master/src/com/android/settings/wifi/p2p/WifiP2pPeer.java, 2011.

[23] "Android Bluetooth Socket," https://developer.android.com/reference/android/bluetooth/BluetoothSocket, 2008.

[24] "Android Bluetooth Server Socket," https://developer.android.com/reference/android/bluetooth/BluetoothServerSocket, 2008.