A Resource-Aware Method for Parallel D2D Data Streaming

International Conference on Computational Collective Intelligence

ICCCI 2021: Computational Collective Intelligence pp 696-707 | Cite as

- Stanisław Saganowski (1) Email author (stanislaw.saganowski@pwr.edu.pl)
- Przemysław Kazienko (1)

1. Department of Artificial Intelligence, Faculty of Information and Communication Technology, Wrocław University of Science and Technology, , Wrocław, Poland

Conference paper
First Online: 30 September 2021

Part of the Lecture Notes in Computer Science book series (LNCS, volume 12876)

Abstract

Smartphones have become an integral part of our lives. One of their crucial functionalities is sharing data. We analyze the communication modules in Android devices (WiFi, Bluetooth, NFC) in terms of parallel data streaming capabilities. We find that increasing the number of concurrent threads reduces the broadcast time, but also consumes a lot of other resources, e.g. RAM. Furthermore, the linear increase in the number of threads created does not guarantee the linear decrease in transfer time. We also find that the Bluetooth 5.0 has, in fact, slower transmission rate than its predecessors.

Keywords

D2D Data transfer Parallel streaming WiFi Direct

This work was partially supported by the National Science Centre, Poland, projects no. 2020/37/B/ST6/03806 and 2016/21/B/ST6/01463; by the statutory funds of the Department of Computational Intelligence, Wroclaw University of Science and Technology; by the European Regional Development Fund as a part of the 2014–2020 Smart Growth Operational Programme, CLARIN - Common Language Resources and Technology Infrastructure, project no. POIR.04.02.00-00C002/19.

1 Introduction

Nowadays, media content plays a crucial role in our life. We share photos and movies, which quality (and file size) grows proportionally to our demands of rapid content transfer. Hence, data transfer technologies are continually being developed. The second-generation 2G cellular networks were commercially launched in 1991, 3G in 1998, and 4G in 2008. Currently, we are desperately waiting for the implementation of the fifth-generation 5G networks, and the requirements stated by the Next Generation Mobile Networks Alliance are tremendous.

A key factor in developing 5G networks is device-to-device (D2D) communication. It allows user devices (UEs) to exchange data directly, without engaging the cellular infrastructure. Researchers and engineers believe that D2D might offload cellular networks, improve spectrum utilization and energy efficiency. Therefore, D2D communication is currently investigated at an unprecedented scale. Every year few surveys are being conducted presenting up-to-date state-of-the-art: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]. Most recently, research is focused on device discovery [13, 14, 15], security and privacy [7, 8, 16], mobility management [17, 18, 19], interference management [20, 21, 22], and power control [23, 24, 25].

All these aspects are fundamental, however, we have to remember that in most cases end users are interested in the highest possible transfer rate with smartphones they already possess and with already deployed technologies. For that reason, we analyze the data transfer modules in nowadays Android devices, namely WiFi, Bluetooth (BT), and Near Field Communication (NFC), in terms of data streaming to many devices at the same time. We investigate whether the attributes such as device system version, BT standard version, the size of the transferred file, and the number of devices to which the file is transferred, influence the transfer time. Based on our analysis, we propose a dynamic parallel broadcast solution that is tailored to the sender resources (the number of CPU cores, available RAM, and overall device efficiency). We validate our solution in a real-world scenario.

The proposed solution will be helpful in many everyday activities like sharing media content, multi-player games, group multi-casting, and local voice services.

2 Related Work

Among existing D2D technologies, the ubiquity makes BT, NFC, and WiFi Direct (WFD) especially attractive. Hence, their critical aspects, such as security and power consumption, are being improved continuously. Recently, Shen et al. [26] introduced a short authentication-string-based key agreement protocol, which guarantees secure WFD connection at the cost of exchanging short password over an out-of-band channel. They successfully integrated the protocol into the existing WFD protocol, and also implemented the solution for Android smartphones. Saleh and Dong [16] proposed secure routing protocols for the vehicular ad hoc networks. Xiao and Li [27] mixed current WFD power saving modes to decrease battery consumption. Usman et al. [28], in turn, achieved savings in power consumption in multi-hop D2D network by controlling the WFD group size and transmit power of the devices.

With the Dynamic Adaptive Streaming over HTTP (DASH) technique, which allows dividing large video files (e.g. live stream) into small chunks, it is possible to utilize D2D collaboration to improve video streaming. In recent years several solutions have been proposed that make use of the cellular D2D and WFD D2D to enhance video streaming [29, 30, 31, 32, 33, 34]. We discuss only the most relevant to our problem.

Gong et al. [29] proposed a cooperative video streaming system in which a group of WFD connected devices exchange chunks of videos between each other. A group of client devices is managed by the group owner (GO), who allocates downloading tasks of clients and broadcasts video chunks among them. Once the client downloads a video chunk through the cellular network, it passes the chunk to the GO, who then sends the chunk to other clients, thus minimizing the number of exchanged packets. The authors validated the system in a real environment and found that a system consisting of three devices is sufficient to ensure smooth video streaming with bitrate more than 10 Mbps.

Le et al. proposed MicroCast [30], a very interesting solution that takes advantage of signal overhearing. The system allows downloading videos over cellular network faster if done in collaboration of devices connected in WiFi Ad-hoc network. At first, the tasks (chunks) are distributed among all participating devices, which then download designated parts. When a single device downloads a chunk it transfers the chunk over a WiFi or BT connection to the device acting as an access point (AP). Other devices in the network obtain the chunk by overhearing; therefore, there is no need to broadcast the chunk by the AP device. To this day, this is the most efficient and the fastest way to share a file with a group of devices, because the chunk is transmitted only once. Unfortunately, the overhearing is not possible on the standard smartphone, unless its operating system (OS) has been modified. For example, to enable overhearing on the Android devices, one has to root the device and install a custom version of the OS. Developing custom OS is not a trivial task, even for an experienced developer. Such a system requires, among others, that only relevant packets from all packets overheard are processed and passed to the application layer. Additionally, rooting smartphones voids their warranty, and the overhearing is highly insecure communication technique.

Jahed et al. [31] presented a scalable cooperative system for multimedia streaming, which is robust against device mobility, allows seamless neighbour discovery and link quality estimation, as well as intelligent clustering and channel allocation algorithms. The system identifies master devices which will function as APs and share content with slave devices. Unfortunately, the system requires a management server that performs optimized group formation, content management, and APs management (channel allocation and power management). The system will not work without the management server, which, due to the complexity of operations, cannot be a smartphone device.

There is also a number of works aiming at reducing: transmission slots, retransmission energy consumption, and the number of retransmission packets in cooperative WFD D2D networks. Proposed solutions utilize the global network topology [35], optimize scheduling decisions [36], and applies integer linear programming and graph partitioning [37, 38, 39, 40].

Although the solutions mentioned above are exciting, they do not fulfil our needs - sharing content with multiple devices simultaneously using explicitly standard (not modified) smartphones. Additionally, most of the proposed solutions are validated on models (e.g. created with MATLAB), in which environmental factors or device imperfections are not considered. In real conditions, a regular TV or fridge can cause a signal disturbance and affect communication. Therefore, our solution has been deployed to standard Android devices and validated in a real-world scenario. Mtibaa et al. [41], have recently performed a study in real conditions, but they focused only on one-to-one communication and the distance between the devices. They arranged three set-ups: (1) indoor line-of-sight, (2) indoor with obstacles (offices), and (3) outdoor line-of-sight, and they considered three connection types: (1) Bluetooth, (2) WiFi ad-hoc, and (3) WiFi Direct. Their findings confirm the intuition that the transfer parameters (e.g. signal strength and the effective throughput) decrease as the distance between devices increases. Interestingly, in the case of line-of-sight indoor set-up, some parameters were preserving high values even with a great distance. The authors explain this as a result of the signal reflections, e.g. in corridors.

To the best of our knowledge, a dynamic parallel broadcast solution proposed in this work is the first one that works with standard devices, do not require any additional devices such as routers or servers, and has been validated in a real-world scenario.

3 Modules for Data Streaming

Most of the nowadays devices are equipped with BT, NFC, and WiFi modules. BT technology is the oldest among them, first presented in 1994 by Ericsson company. It was designed to connect devices located close to each other wirelessly, the main application idea was to connect phones with wireless headphones or hands-free sets. BT utilizes 2402–2480 MHz frequency range and since its debut uses frequency-hopping spread spectrum (FHSS). BT protocol implements master-slave architecture, thus allowing the master for up to 7 connections with slave devices. Most Android devices on the market have BT version 4.0, 4.1 or 4.2, which, according to manufacturers, performs with the maximum data rate of 24 Mbps and its maximum range is 100 m. In 2016 the BT version 5.0 was introduced, advertised with a boost of the data rate to 48 Mbps, and a more excellent range of up to 400 m. In 2018 the first smartphones with BT 5.0 were released. Since 2010 BT Low Energy is also available, a version featuring much less energy consumption and maintaining similar communication parameters. However, since it favours low power consumption over the file transfer speed, it is not considered in this study.

Not long after BT release the IEEE 802.11 (WiFi) was standardized and deployed as well. WiFi was designed to allow wireless communication at high speed from the very beginning, starting at 2 Mbps in 1997 to 9608 Mbps available with the most recent, 6th generation WiFi. The most common WiFi standards implemented in nowadays smartphones are 802.11 a, b, g, n, and ac, which allows communicating at 2.4 and 5 GHz radio frequencies. The WiFi communication at 5 GHz band allows using more channels, which results in better transfer rate. Additionally, the number of devices supporting 5 GHz band is significantly lower. Thus the interference is not as bad as among 2.4 GHz band devices. The only downside of the 5 GHz band is that it covers a much shorter range.

Two WiFi modes are particularly interesting for our study: (1) WiFi Direct, and (2) WiFi ad-hoc. Devices operating in WiFi ad-hoc mode can communicate directly with each other, without the need to talk to an AP. Each device is required to route traffic, hence creating a decentralized network properly. Unfortunately, the ad-hoc mode is not available on standard Android devices. Therefore it is not considered in this study. On the contrary, all devices running Android version 4.0 or higher can use WiFi Direct (WFD). In this mode, one of the devices acts as an AP and routes all the traffic. However, no additional router or access point is necessary. The standard does not specify the maximum number of devices that can join the WFD network. However, the unmodified Android implementation assumes 254 IP addresses to be assigned to clients (besides the AP address). This would result in up to 255 devices connected to the network, but phone manufacturers usually limit this number to 8–12 devices, besides the device acting as an AP.

As the name states, Near Field Communication (NFC) technology, transmits the data within the short-range only, up to 20 cm. It operates at 13.56 MHz radiofrequency and data rate of up to 424 kbps. The NFC technology, available on Android device, can work in three modes: (1) to read from and write to NFC tags, that do not require power (the slight current generated by the inductive coupling allows to exchange the information), (2) to respond to the card terminal, thus acting as an intelligent card, and (3) peer-to-peer mode allows to transfer data between two devices. Our use-case assumes the communication among multiple devices, which NFC cannot provide (at least not in parallel). Therefore, the NFC technology will be used only as a reference and possible way of communication when there are only two devices.

Let us emphasize that all technologies mentioned above, namely Bluetooth, WiFi Direct, and NFC, are available on standard Android devices since system version 2.0, 4.0, and 2.3 respectively (for comparison the most recent Android version is 10.0). Hence BT, WFD and NFC technologies are available on 99.9% of devices present in the market.

4 Resource-Aware Parallel Data Streaming Method

The most straightforward approach to transfer data to multiple devices is the serial transfer. It means sending data to the first recipient, waiting for confirmation of receipt, and then the analogous data transfer to other recipients. However, if the data is static and does not need to be updated, we may attempt transferring data in parallel. To achieve this task on the Android platform, we have used Java multi-threading, which requires extending the Thread class and implementing the Runnable interface.

The method assumes that the sender and recipients are already configured (paired and connected). At first, resources (the number of CPU cores and RAM) available on the sender device are identified. The sender device is responsible for maintaining all the connections with recipients (clients). Next, depending on the number of recipients and available resources, concurrent threads are dynamically created. If the resources are sufficient enough, a separate thread is created for each client. If this is not possible, recipients not assigned to the thread are waiting in the queue until a thread is available. Finally, the data is transferred in parallel in all threads until the recipient pool is exhausted.

In the case when there is only one recipient, the transfer is handled in the main thread of the application (no additional threads are created), because of the overhead cost of creating the thread (more details on this topic are provided in the experimental section). The code of the application (and method) is publicly available at [42].

5 Experimental Setup

The experiments were conducted using nine commonly available smartphones, see Table 1. The devices were selected in such a way to cover various market segments. We have used both, top models (e.g. Samsung Galaxy S9+), as well as less expensive models (e.g. LG G2 mini). This very well reflects the diversity of devices on the market, and also allows us to analyze various parameters, e.g. the BT version, the number of CPU cores, and other. For all devices, the latest operating system versions and updates were installed.

All the experiments were performed using the Android application, which code is publicly available at [42]. The application allows to set-up a connection between sender and recipients using four considered technologies (BT, WiFi, NFC, Amazon Web Services - AWS). Furthermore, the application allows to transfer prepared files and measure the time of data transfer. Our research is focused on the time required to transfer data from the sender device to the receiver devices. We consider from one, up to eight concurrent recipients (clients). The time measured in the study is related to the file transfer phase only, i.e. we assume that the devices are already paired and connected. The timer embedded into the application starts counting the time when the transfer to the first recipient begins and ends when the last recipient notifies about the successful file transmission. Each experiment (file streaming) is repeated 30 times, and the data transfer time is averaged.

We are considering four data transfer technologies: BT, WiFi, NFC, and, for reference purposes, Amazon Web Services (AWS) cloud transfer. When analyzing the data transfer time, we use various file sizes: 10 kb (the size of short text message with emoticons), 100 kb, 300 kb (the size of a compressed photos sent through Messenger), 500 kb, 1 Mb, 2 Mb (the size of a compressed short movies sent through Messenger), 5 Mb, 10 Mb (the size of a RAW photo or a short video sent through WhatsApp). We do not analyze device discovery, pairing, nor connecting time, as it usually takes a fraction of the data transfer time, and is performed once per session (connecting devices) or once per lifetime (pairing devices). Furthermore, ping and the packet loss ratio was close to zero due to the short distance, and are not considered as well.

Table 1.

The specification of devices used in the experiment.

Device name	Full name	Version	Release date	CPU [cores]		RAM [GB]	WiFi [GHz]	BT version	OS version
Mate20	Huawei Mate 20 Pro		2018, Nov	8	2 × 2,6 2 × 1,92 4 × 1,8	6	5, 2.4 a/b/g/n/ac	5.0	9.0
S9	Samsung Galaxy S9+	G965F	2018, Mar	8	4 × 22,7 4 × 21,8	6	5, 2.4 a/b/g/n/ac	5.0	8.0
Mate9	Huawei Mate 9		2016, Dec	8	4 × 22,4 4 × 21,8	4	5, 2.4 a/b/g/n/ac	4.2	8.0
S6	Samsung Galaxy S6 edge+	G928F	2015, Aug	8	4 × 22,1 4 × 21,5	4	5, 2.4 a/b/g/n/ac	4.2	7.0
Note4	Samsung Galaxy Note 4	N910F	2014, Oct	4	4 × 22,7	3	5, 2.4 a/b/g/n/ac	4.1	6.0.1
Note4	Samsung Galaxy Note 4	N910F	2014, Oct	4	4 × 22,7	3	5, 2.4 a/b/g/n/ac	4.1	6.0.1
G3	LG G3	D855	2014, Jun	4	4 × 22.5	2	5, 2.4 a/b/g/n/ac	4.0	6.0
G2	LG G2 mini		2014, Apr	4	4 × 21,2	1	2.4 b/g/n	4.0	5.0.2
Note2	Samsung Galaxy Note II	N7100	2012, Sep	4	4 × 21,6	2	2.4 a/b/g/n	4.0	4.4.2

6 Experimental Results

At first, we have compared the time of file transmission to a single recipient using four considered technologies (BT, WiFi, NFC, and AWS), see Fig. 1A. The results support the intuition that the transfer time is correlated linearly with the file size. By far the fastest technology is WiFi - on average it took only 17 ms for a 10 kb file transfer, and about 3 s for a 10 Mb transmission. At the same time, NFC required about 8 s for 10 kb, and 80 s for 10 Mb, while BT needed 131 ms for 10 kb and 135 s for a 10 Mb file. The streaming via the AWS cloud was the slowest and required 540 ms for 10 kb and 150 s for a 10 Mb transfer. Interestingly, NFC technology is slower than BT and AWS for files up to 1 Mb, but faster for data larger than 1 Mb. Perhaps the overhead of initializing the NFC communication is more substantial than for BT. For BT, WiFi, and AWS, we have also checked the transfer time in the parallel mode (dashed lines in Fig. 1A) - of course, only one thread was used, as the file has been transferred to a single device. For all considered technologies, one can observe the time overhead for creating a separate thread for file transmission.

The advantage of parallel streaming quickly becomes apparent when we share data with more than one device. See Fig. 1B for the comparison of BT, WiFi, and AWS when transferring a file to four devices. For each technology, the parallel transfer is much faster than serial transfer. For WiFi, the time saving is as much as 33%, for BT 27% and AWS 14%. The advantage of parallel streaming over serial streaming is even more obvious in Fig. 1C, where we analyze file transmission using WiFi serial and WiFi parallel methods to different numbers of devices. In this case, the number of parallel threads is equal to the number of recipients. For better readability, only the results of 1 Mb and 10 Mb transfers are presented. For two recipients, the parallel transfer is faster than serial transfer by about 20%. When sharing data with eight recipients, the parallel transfer is two times faster than serial streaming.

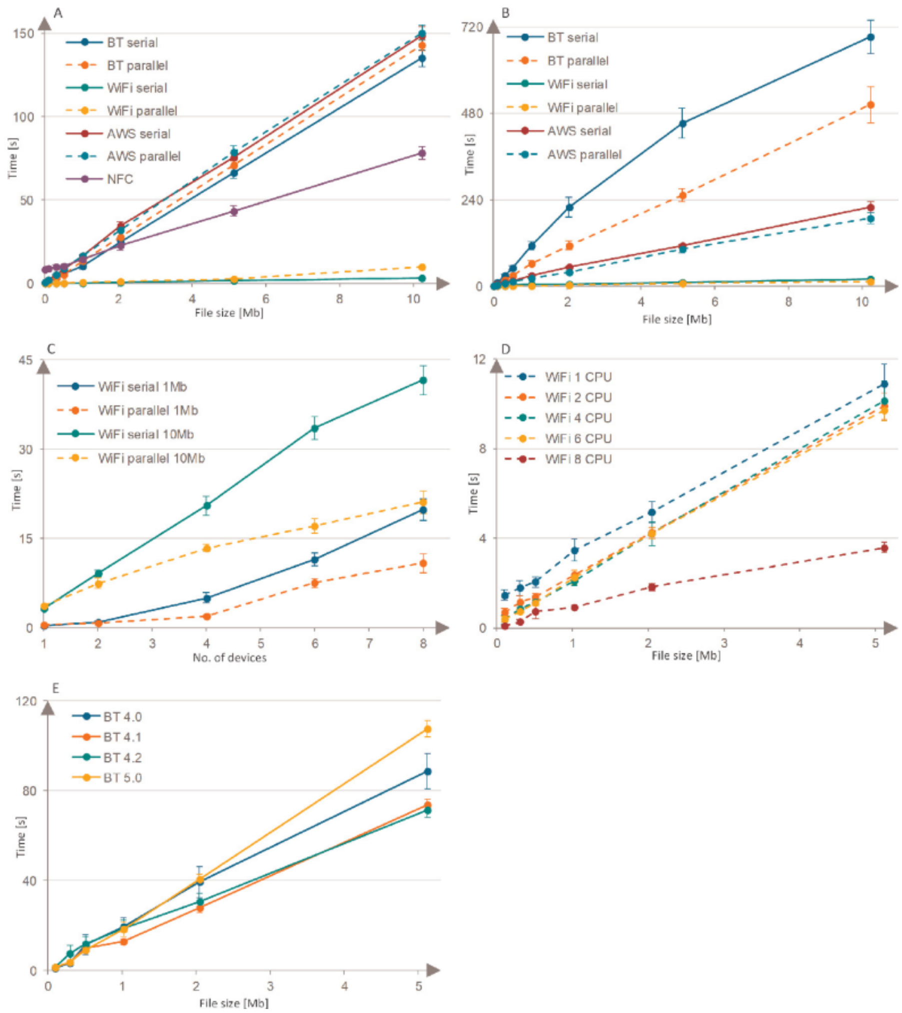

Fig. 1.

(**A**) Data transfer time to a single device. (**B**) Data transfer time to four devices. (**C**) Comparison of WiFi serial and parallel transfers. (**D**) Influence of the number of threads used in the data transfer. (**E**) Data transfer time for various Bluetooth versions.

At this point, the obvious question is why parallel data transfer to eight devices is only twice as fast as the serial transfer. We would expect to reduce the time by almost eight times, as we use eight parallel threads. It turns out that increasing the number of threads leads to much faster consumption of other resources, i.e. RAM. When the RAM is running out, it needs to be released. Because Android uses the Java run time environment, releasing RAM is done automatically through the Garbage Collector (GC) process. GC can even pause all working threads to perform some critical work, e.g. moving variables. This situation occurs much more often during the transmission of large amounts of data and when using multiple threads. Furthermore, data transfer technology also has an impact on resource consumption. For example, WiFi typically consumes over 40% of CPU resources, while BT consumes at maximum 15% of CPU.

To validate whether the time profit of the parallel transfer is impacted by time overhead caused by overfilled RAM, we have conducted another experiment. The data has been transmitted to eight devices several times, and each time the number of working threads has been reduced (from eight threads to a single thread), see Fig. 1D. It turns out that despite a much higher number of GC interventions, the parallel transfer with as many threads as recipients is the most time-effective. In this case, the transfer time using two, four, and six threads do not differ significantly. However, the transfer using eight threads is almost three times faster than the transmission on a single thread.

The next parameter tested was the BT protocol version. We have compared versions 4.0, 4.1, 4.2, and 5.0 see Fig. 1E. Surprisingly, BT 5.0 performed the worst out of all considered BT versions. Let us remind that BT 5.0 is supposed to offer twice the speed than BT 4.2. Therefore, we have repeated the experiment twice to make sure that nothing random disturbed the transfer, e.g. system updates or application synchronization running in the background. Both experiments (each consisting of 30 repetitions) gave the same result. Our only assumption is that one of the manufacturers incorrectly implemented the standard, hence the result worse than BT 4.2. To validate that experiments will be repeated with devices of different manufacturers. Other results are in line with expectations. BT versions 4.1 and 4.2 were better than version 4.0; however, it may as well be related to the device's parameters other than the BT version. Devices with a higher BT version are newer and have more RAM and CPU power.

7 Conclusion and Future Work

We have analyzed the data transfer time between multiple devices using technologies embedded in the standard smartphones, namely WiFi, Bluetooth (BT), and Near Field Communication (NFC). Let us emphasize that we have used standard, non-modified, smartphones, thus our research applies to 99.9% of devices on the market. The experiments conducted in a real-world scenario suggest that sharing content with more than one device should be performed using parallel WiFi connection. Furthermore, the sender device should use as many threads as recipients awaiting the data, if such resources are available. However, the linear increase in the number of threads created does not guarantee the linear decrease in transfer time, unless the RAM resources are increased as well. The Garbage Collector process responsible for releasing the RAM resources has a significant impact on the file transmission time. When transmitting the data to a single device, it is better to use the main thread of the application, since the separate thread would create the overhead. The experiments also confirmed that the transfer time is correlated linearly with the file size being transmitted and with the number of recipients. Last but not least, BT 5.0 performed much worse than expected, transferring files slower than its predecessors (BT 4.2, BT 4.1, and even BT 4.0).

The next possible research direction is reducing the number of GC calls (e.g. by sharing memory between threads), which would further increase the parallel transfer advantage over the serial file transfer. Nevertheless, another parameter interesting to analyze is the WiFi band, i.e. 2.4 GHz vs 5 GHz.

References

1. Camps-Mur, D., Garcia-Saavedra, A., Serrano, P.: Device-to-device communications with Wi-Fi Direct: overview and experimentation. IEEE Wirel. Commun. **20**(3), 96–104 (2013)
CrossRef (https://doi.org/10.1109/MWC.2013.6549288)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Device-to-device%20communications%20with%20Wi-Fi%20Direct%3A%20overview%20and%20experimentation&author=D.%20Camps-Mur&author=A.%20Garcia-Saavedra&author=P.%20Serrano&journal=IEEE%20Wirel.%20Commun.&volume=20&issue=3&pages=96-104&publication_year=2013)

2. Asadi, A., Wang, Q., Mancuso, V.: A survey on device-to-device communication in cellular networks. IEEE Commun. Surv. Tutor. **16**(4), 1801–1819 (2014)
CrossRef (https://doi.org/10.1109/COMST.2014.2319555)
Google Scholar (http://scholar.google.com/scholar_lookup?title=A%20survey%20on%20device-to-device%20communication%20in%20cellular%20networks&author=A.%20Asadi&author=Q.%20Wang&author=V.%20Mancuso&journal=IEEE%20Commun.%20Surv.%20Tutor.&volume=16&issue=4&pages=1801-1819&publication_year=2014)

3. Feng, D., Lu, L., et al.: Device-to-device communications in cellular networks. IEEE Comm. Mag. **52**(4), 49–55 (2014)
CrossRef (https://doi.org/10.1109/MCOM.2014.6807946)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Device-to-device%20communications%20in%20cellular%20networks&author=D.%20Feng&author=L.%20Lu&journal=IEEE%20Comm.%20Mag.&volume=52&issue=4&pages=49-55&publication_year=2014)

4. Liu, J., Kato, N., et al.: Device-to-device communication in LTE-advanced networks: a survey. IEEE Commun. Surv. Tutor. **17**(4), 1923–1940 (2015)
CrossRef (https://doi.org/10.1109/COMST.2014.2375934)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Device-to-device%20communication%20in%20LTE-advanced%20networks%3A%20a%20survey&author=J.%20Liu&author=N.%20Kato&journal=IEEE%20Commun.%20Surv.%20Tutor.&volume=17&issue=4&pages=1923-1940&publication_year=2015)

5. Noura, M., Nordin, R.: A survey on interference management for device-to-device (D2D) communication and its challenges in 5G networks. J. Netw. Comput. Appl. **71**, 130–150 (2016)
CrossRef (https://doi.org/10.1016/j.jnca.2016.04.021)
Google Scholar (http://scholar.google.com/scholar_lookup?title=A%20survey%20on%20interference%20management%20for%20device-to-device%20%28D2D%29%20communication%20and%20its%20challenges%20in%205G%20networks&author=M.%20Noura&author=R.%20Nordin&journal=J.%20Netw.%20Comput.%20Appl.&volume=71&pages=130-150&publication_year=2016)

6. Gandotra, P., Jha, R.K.: Device-to-device communication in cellular networks: a survey. J. Netw. Comput. Appl. **71**, 99–117 (2016)
CrossRef (https://doi.org/10.1016/j.jnca.2016.06.004)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Device-to-device%20communication%20in%20cellular%20networks%3A%20a%20survey&author=P.%20Gandotra&author=RK.%20Jha&journal=J.%20Netw.%20Comput.%20Appl.&volume=71&pages=99-117&publication_year=2016)

7. Haus, M., Waqas, M., et al.: Security and privacy in device-to-device (D2D) communication: a review. IEEE Commun. Surv. Tutor. **19**(2), 1054–1079 (2017)
CrossRef (https://doi.org/10.1109/COMST.2017.2649687)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Security%20and%20privacy%20in%20device-to-device%20%28D2D%29%20communication%3A%20a%20review&author=M.%20Haus&author=M.%20Waqas&journal=IEEE%20Commun.%20Surv.%20Tutor.&volume=19&issue=2&pages=1054-1079&publication_year=2017)

8. Hamoud, O.N., Kenaza, T., Challal, Y.: Security in device-to-device communications: a survey. IET Netw. **7**(1), 14–22 (2017)
CrossRef (https://doi.org/10.1049/iet-net.2017.0119)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Security%20in%20device-to-device%20communications%3A%20a%20survey&author=ON.%20Hamoud&author=T.%20Kenaza&author=Y.%20Challal&journal=IET%20Netw.&volume=7&issue=1&pages=14-22&publication_year=2017)

9. Wang, M., Yan, Z.: A survey on security in D2D communications. Mob. Netw. Appl. **22**(2), 195–208 (2017)
CrossRef (https://doi.org/10.1007/s11036-016-0741-5)
Google Scholar (http://scholar.google.com/scholar_lookup?title=A%20survey%20on%20security%20in%20D2D%20communications&author=M.%20Wang&author=Z.%20Yan&journal=Mob.%20Netw.%20Appl.&volume=22&issue=2&pages=195-208&publication_year=2017)

10. Fomichev, M., Álvarez, F., et al.: Survey and systematization of secure device pairing. IEEE Commun. Surv. Tutor. **20**(1), 517–550 (2017)
CrossRef (https://doi.org/10.1109/COMST.2017.2748278)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Survey%20and%20systematization%20of%20secure%20device%20pairing&author=M.%20Fomichev&author=F.%20%C3%81lvarez&journal=IEEE%20Commun.%20Surv.%20Tutor.&volume=20&issue=1&pages=517-550&publication_year=2017)

11. Jameel, F., Hamid, Z., et al.: A survey of device-to-device communications: research issues and challenges. IEEE Commun. Surv. Tutor. **20**, 2133–2168 (2018)
CrossRef (https://doi.org/10.1109/COMST.2018.2828120)
Google Scholar (http://scholar.google.com/scholar_lookup?title=A%20survey%20of%20device-to-device%20communications%3A%20research%20issues%20and%20challenges&author=F.%20Jameel&author=Z.%20Hamid&journal=IEEE%20Commun.%20Surv.%20Tutor.&volume=20&pages=2133-2168&publication_year=2018)

12. Shaikh, F.S., Wismüller, R.: Routing in multi-hop cellular device-to-device (D2D) networks: a survey. IEEE Commun. Surv. Tutor. **20**, 2622–2657 (2018)
CrossRef (https://doi.org/10.1109/COMST.2018.2848108)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Routing%20in%20multi-hop%20cellular%20device-to-device%20%28D2D%29%20networks%3A%20a%20survey&author=FS.%20Shaikh&author=R.%20Wism%C3%BCller&journal=IEEE%20Commun.%20Surv.%20Tutor.&volume=20&pages=2622-2657&publication_year=2018)

13. Li, H.-B., Miura, R., Kojima, F.: Channel access proposal for enabling quick discovery for D2D wireless networks. In: International Conference on Computing, Networking and Communications (ICNC), pp. 1012–1016. IEEE (2017)
Google Scholar (https://scholar.google.com/scholar?q=Li%2C%20H.-B.%2C%20Miura%2C%20R.%2C%20Kojima%2C%20F.%3A%20Channel%20access%20proposal%20for%20enabling%20quick%20discovery%20for%20D2D%20wireless%20networks.%20In%3A%20International%20Conference%20on%20Computing%2C%20Networking%20and%20Communications%20%28ICNC%29%2C%20pp.%201012%E2%80%931016.%20IEEE%20%282017%29)

14. Xu, C., Gao, C., et al.: Social network-based content delivery in device-to-device underlay cellular networks using matching theory. IEEE Access **5**, 924–937 (2017)
CrossRef (https://doi.org/10.1109/ACCESS.2016.2621010)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Social%20network-based%20content%20delivery%20in%20device-to-device%20underlay%20cellular%20networks%20using%20matching%20theory&author=C.%20Xu&author=C.%20Gao&journal=IEEE%20Access&volume=5&pages=924-937&publication_year=2017)

15. Kushalad, K., Sarkar, M., Patel, P.: Asynchronous device discovery and rendezvous protocol for D2D communication. In: Conference on Computer Communications Workshops, pp. 199–200. IEEE (2016)
Google Scholar (https://scholar.google.com/scholar?q=Kushalad%2C%20K.%2C%20Sarkar%2C%20M.%2C%20Patel%2C%20P.%3A%20Asynchronous%20device%20discovery%20and%20rendezvous%20protocol%20for%20D2D%20communication.%20In%3A%20Conference%20on%20Computer%20Communications%20Workshops%2C%20pp.%20199%E2%80%93200.%20IEEE%20%282016%29)

16. Saleh, M., Dong, L.: Secure location-aided routing protocols with Wi-Fi Direct for vehicular ad hoc networks. arXiv preprint arXiv:1707.00654 (http://arxiv.org/abs/1707.00654) (2017)

17. Wang, Z., Sun, L., Zhang, M., Pang, H., Tian, E., Zhu, W.: Propagation-and mobility-aware D2D social content replication. IEEE Trans. Mob. Comput. **16**(4), 1107–1120 (2017)
CrossRef (https://doi.org/10.1109/TMC.2016.2582159)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Propagation-and%20mobility-aware%20D2D%20social%20content%20replication&author=Z.%20Wang&author=L.%20Sun&author=M.%20Zhang&author=H.%20Pang&author=E.%20Tian&author=W.%20Zhu&journal=IEEE%20Trans.%20Mob.%20Comput.&volume=16&issue=4&pages=1107-1120&publication_year=2017)

18. Wang, R., Zhang, J., Song, S., Letaief, K.B.: Mobility-aware caching in D2D networks. IEEE Trans. Wirel. Commun. **16**(8), 5001–5015 (2017)
CrossRef (https://doi.org/10.1109/TWC.2017.2705038)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Mobility-aware%20caching%20in%20D2D%20networks&author=R.%20Wang&author=J.%20Zhang&author=S.%20Song&author=KB.%20Letaief&journal=IEEE%20Trans.%20Wirel.%20Commun.&volume=16&issue=8&pages=5001-5015&publication_year=2017)

19. Krishnan, S., Dhillon, H.S.: Effect of user mobility on the performance of device-to-device networks with distributed caching. IEEE Wirel. Commun. Lett. **6**(2), 194–197 (2017)
CrossRef (https://doi.org/10.1109/LWC.2017.2654455)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Effect%20of%20user%20mobility%20on%20the%20performance%20of%20device-to-device%20networks%20with%20distributed%20caching&author=S.%20Krishnan&author=HS.%20Dhillon&journal=IEEE%20Wirel.%20Commun.%20Lett.&volume=6&issue=2&pages=194-197&publication_year=2017)

20. Yang, T., Cheng, X., Shen, X., Chen, S., Yang, L.: QoS-aware interference management for vehicular D2D relay networks. J. Commun. Inf. Netw. **2**(2), 75–90 (2017)
CrossRef (https://doi.org/10.1007/s41650-017-0026-6)
Google Scholar (http://scholar.google.com/scholar_lookup?title=QoS-aware%20interference%20management%20for%20vehicular%20D2D%20relay%20networks&author=T.%20Yang&author=X.%20Cheng&author=X.%20Shen&author=S.%20Chen&author=L.%20Yang&journal=J.%20Commun.%20Inf.%20Netw.&volume=2&issue=2&pages=75-90&publication_year=2017)

21. Celik, A., Radaydeh, R.M.M., Al-Qahtani, F.S., Alouini, M.-S.: Resource allocation and interference management for D2D-enabled DL/UL decoupled het-nets. IEEE Access **5**, 22735–22749 (2017)
CrossRef (https://doi.org/10.1109/ACCESS.2017.2760350)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Resource%20allocation%20and%20interference%20management%20for%20D2D-enabled%20DL%2FUL%20decoupled%20het-nets&author=A.%20Celik&author=RMM.%20Radaydeh&author=FS.%20Al-Qahtani&author=M-S.%20Alouini&journal=IEEE%20Access&volume=5&pages=22735-22749&publication_year=2017)

22. Sun, W., Yuan, D., et al.: Cluster-based radio resource management for D2D-supported safety-critical V2X communications. IEEE Trans. Wirel. Commun. **15**(4), 2756–2769 (2016)
MathSciNet (http://www.ams.org/mathscinet-getitem?mr=3497080)
CrossRef (https://doi.org/10.1109/TWC.2015.2509978)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Cluster-based%20radio%20resource%20management%20for%20D2D-supported%20safety-critical%20V2X%20communications&author=W.%20Sun&author=D.%20Yuan&j

ournal=IEEE%20Trans.%20Wirel.%20Commun.&volume=15&issue=4&pages=2756-2769&publication_year=2016)

23. Xu, H., Huang, N., Yang, Z., Shi, J., Wu, B., Chen, M.: Pilot allocation and power control in D2D underlay massive MIMO systems. IEEE Commun. Lett. **21**(1), 112–115 (2017)
CrossRef (https://doi.org/10.1109/LCOMM.2016.2597236)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Pilot%20allocation%20and%20power%20control%20in%20D2D%20underlay%20massive%20MIMO%20systems&author=H.%20Xu&author=N.%20Huang&author=Z.%20Yang&author=J.%20Shi&author=B.%20Wu&author=M.%20Chen&journal=IEEE%20Commun.%20Lett.&volume=21&issue=1&pages=112-115&publication_year=2017)

24. Yang, C., Li, J., et al.: Distributed interference and energy-aware power control for ultra-dense D2D networks: a mean field game. IEEE Trans. Wirel. Commun. **16**(2), 1205–1217 (2017)
CrossRef (https://doi.org/10.1109/TWC.2016.2641959)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Distributed%20interference%20and%20energy-aware%20power%20control%20for%20ultra-dense%20D2D%20networks%3A%20a%20mean%20field%20game&author=C.%20Yang&author=J.%20Li&journal=IEEE%20Trans.%20Wirel.%20Commun.&volume=16&issue=2&pages=1205-1217&publication_year=2017)

25. Zhai, D., Sheng, M., et al.: Energy-saving resource management for D2D and cellular coexisting networks enhanced by hybrid multiple access technologies. IEEE Trans. Wirel. Commun. **16**(4), 2678–2692 (2017)
CrossRef (https://doi.org/10.1109/TWC.2017.2671863)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Energy-saving%20resource%20management%20for%20D2D%20and%20cellular%20coexisting%20networks%20enhanced%20by%20hybrid%20multiple%20access%20technologies&author=D.%20Zhai&author=M.%20Sheng&journal=IEEE%20Trans.%20Wirel.%20Commun.&volume=16&issue=4&pages=2678-2692&publication_year=2017)

26. Shen, W., Yin, B., et al.: Secure device-to-device communications over WiFi direct. IEEE Netw. **30**(5), 4–9 (2016)
CrossRef (https://doi.org/10.1109/MNET.2016.7579020)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Secure%20device-to-device%20communications%20over%20WiFi%20direct&author=W.%20Shen&author=B.%20Yin&journal=IEEE%20Netw.&volume=30&issue=5&pages=4-9&publication_year=2016)

27. Xiao, S., Li, W.: Mixed power saving mode for D2D communications with Wi-Fi Direct. In: Qiao, F., Patnaik, S., Wang, J. (eds.) ICMIR 2017. AISC, vol. 691, pp. 234–240. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-70990-1_35 (https://doi.org/10.1007/978-3-319-70990-1_35)
CrossRef (https://doi.org/10.1007/978-3-319-70990-1_35)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Mixed%20power%20saving%20mode%20for%20D2D%20communications%20with%20Wi-

Fi%20Direct&author=S.%20Xiao&author=W.%20Li&pages=234-240&publication_year=2018)

28. Usman, M., Asghar, M.R., et al.: Towards energy efficient multi-hop D2D networks using WiFi Direct. In: IEEE Global Communications Conference, pp. 1–7. IEEE (2017)
Google Scholar (https://scholar.google.com/scholar?q=Usman%2C%20M.%2C%20Asghar%2C%20M.R.%2C%20et%20al.%3A%20Towards%20energy%20efficient%20multi-hop%20D2D%20networks%20using%20WiFi%20Direct.%20In%3A%20IEEE%20Global%20Communications%20Conference%2C%20pp.%201%E2%80%937.%20IEEE%20%282017%29)

29. Gong, Q., Guo, Y., Chen, Y., Liu, Y., Xie, F.: Design and evaluation of a WiFi-Direct based LTE cooperative video streaming system. In: Global Communications Conference (GLOBECOM), pp. 1–6. IEEE (2016)
Google Scholar (https://scholar.google.com/scholar?q=Gong%2C%20Q.%2C%20Guo%2C%20Y.%2C%20Chen%2C%20Y.%2C%20Liu%2C%20Y.%2C%20Xie%2C%20F.%3A%20Design%20and%20evaluation%20of%20a%20WiFi-Direct%20based%20LTE%20cooperative%20video%20streaming%20system.%20In%3A%20Global%20Communications%20Conference%20%28GLOBECOM%29%2C%20pp.%201%E2%80%936.%20IEEE%20%282016%29)

30. Le, A., Keller, L., et al.: MicroCast: cooperative video streaming using cellular and local connections. IEEE/ACM Trans. Netw. **24**(5), 2983–2999 (2016)
CrossRef (https://doi.org/10.1109/TNET.2015.2501349)
Google Scholar (http://scholar.google.com/scholar_lookup?title=MicroCast%3A%20cooperative%20video%20streaming%20using%20cellular%20and%20local%20connections&author=A.%20Le&author=L.%20Keller&journal=IEEE%2FACM%20Trans.%20Netw.&volume=24&issue=5&pages=2983-2999&publication_year=2016)

31. Jahed, K., Sharafeddine, S., et al.: Scalable multimedia streaming in wireless networks with device-to-device cooperation. In: Proceedings of the 2016 ACM on Multimedia Conference, pp. 762–764. ACM (2016)
Google Scholar (https://scholar.google.com/scholar?q=Jahed%2C%20K.%2C%20Sharafeddine%2C%20S.%2C%20et%20al.%3A%20Scalable%20multimedia%20streaming%20in%20wireless%20networks%20with%20device-to-device%20cooperation.%20In%3A%20Proceedings%20of%20the%202016%20ACM%20on%20Multimedia%20Conference%2C%20pp.%20762%E2%80%93764.%20ACM%20%282016%29)

32. Wainer, G., Fernandes, S., et al.: Improving video streaming over cellular networks with dash-based device-to-device streaming. In: Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS), pp. 1–8. IEEE (2017)
Google Scholar (https://scholar.google.com/scholar?q=Wainer%2C%20G.%2C%20Fernandes%2C%20S.%2C%20et%20al.%3A%20Improving%20video%20streaming%20over%20cellular%20networks%20with%20dash-based%20device-to-device%20streaming.%20In%3A%20Symposium%20on%20Performance%20Eva

luation%20of%20Computer%20and%20Telecommunication%20Systems%20%28SPECTS%29%2C%20pp.%201%E2%80%938.%20IEEE%20%282017%29)

33. Fernandes, S., Wainer, G., et al.: Dash-based peer-to-peer video streaming in cellular networks. In: International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS), pp. 1–8. IEEE (2016)
Google Scholar (https://scholar.google.com/scholar?q=Fernandes%2C%20S.%2C%20Wainer%2C%20G.%2C%20et%20al.%3A%20Dash-based%20peer-to-peer%20video%20streaming%20in%20cellular%20networks.%20In%3A%20International%20Symposium%20on%20Performance%20Evaluation%20of%20Computer%20and%20Telecommunication%20Systems%20%28SPECTS%29%2C%20pp.%201%E2%80%938.%20IEEE%20%282016%29)

34. Siris, V.A., Dimopoulos, D.: Multi-source mobile video streaming with proactive caching and D2D communication. In: IEEE 16th International Symposium on World of Wireless, Mobile and Multimedia Networks (WoWMoM), pp. 1–6. IEEE (2015)
Google Scholar (https://scholar.google.com/scholar?q=Siris%2C%20V.A.%2C%20Dimopoulos%2C%20D.%3A%20Multi-source%20mobile%20video%20streaming%20with%20proactive%20caching%20and%20D2D%20communication.%20In%3A%20IEEE%2016th%20International%20Symposium%20on%20World%20of%20Wireless%2C%20Mobile%20and%20Multimedia%20Networks%20%28WoWMoM%29%2C%20pp.%201%E2%80%936.%20IEEE%20%282015%29)

35. Chen, Y.-J., Wang, L.-C., Wang, K., Ho, W.-L.: Topology-aware network coding for wireless multicast. IEEE Syst. J. **12**, 3683–3692 (2018)
CrossRef (https://doi.org/10.1109/JSYST.2018.2802493)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Topology-aware%20network%20coding%20for%20wireless%20multicast&author=Y-J.%20Chen&author=L-C.%20Wang&author=K.%20Wang&author=W-L.%20Ho&journal=IEEE%20Syst.%20J.&volume=12&pages=3683-3692&publication_year=2018)

36. Zhan, C., Wen, Z.: Efficient video transmission in D2D assisted mobile cloud networks. China Commun. **13**(8), 74–83 (2016)
CrossRef (https://doi.org/10.1109/CC.2016.7563690)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Efficient%20video%20transmission%20in%20D2D%20assisted%20mobile%20cloud%20networks&author=C.%20Zhan&author=Z.%20Wen&journal=China%20Commun.&volume=13&issue=8&pages=74-83&publication_year=2016)

37. Zhan, C., Wen, Z., Wang, X., Zhu, L.: Device-to-device assisted wireless video delivery with network coding. Ad Hoc Netw. **69**, 76–85 (2018)
CrossRef (https://doi.org/10.1016/j.adhoc.2017.11.001)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Device-to-device%20assisted%20wireless%20video%20delivery%20with%20network%20coding&author=C.%20Zhan&author=Z.%20Wen&author=X.%20Wang&author=L.%20Zhu&journal=Ad%20Hoc%20Netw.&volume=69&pages=76-85&publication_year=2018)

38. Chi, K., Yu, Z., Li, Y., Zhu, Y.-H.: Energy-efficient D2D communication based retransmission scheme for reliable multicast in wireless cellular network. IEEE

Access **6**, 31469–31480 (2018)
CrossRef (https://doi.org/10.1109/ACCESS.2018.2842065)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Energy-efficient%20D2D%20communication%20based%20retransmission%20scheme%20for%20reliable%20multicast%20in%20wireless%20cellular%20network&author=K.%20Chi&author=Z.%20Yu&author=Y.%20Li&author=Y-H.%20Zhu&journal=IEEE%20Access&volume=6&pages=31469-31480&publication_year=2018)

39. Xia, Z., Yan, J., Liu, Y.: Cooperative content delivery in multicast multihop device-to-device networks. IEEE Access **5**, 6314–6324 (2017)
CrossRef (https://doi.org/10.1109/ACCESS.2017.2672996)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Cooperative%20content%20delivery%20in%20multicast%20multihop%20device-to-device%20networks&author=Z.%20Xia&author=J.%20Yan&author=Y.%20Liu&journal=IEEE%20Access&volume=5&pages=6314-6324&publication_year=2017)

40. Militano, L., Condoluci, M., et al.: Single frequency-based device-to-device-enhanced video delivery for evolved multimedia broadcast and multicast services. IEEE Trans. Broadcast. **61**(2), 263–278 (2015)
CrossRef (https://doi.org/10.1109/TBC.2015.2400824)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Single%20frequency-based%20device-to-device-enhanced%20video%20delivery%20for%20evolved%20multimedia%20broadcast%20and%20multicast%20services&author=L.%20Militano&author=M.%20Condoluci&journal=IEEE%20Trans.%20Broadcast.&volume=61&issue=2&pages=263-278&publication_year=2015)

41. Mtibaa, A., Emam, A., et al.: On practical device-to-device wireless communication: a measurement driven study. In: Wireless Communications and Mobile Computing Conference, pp. 409–414. IEEE (2017)
Google Scholar (https://scholar.google.com/scholar?q=Mtibaa%2C%20A.%2C%20Emam%2C%20A.%2C%20et%20al.%3A%20On%20practical%20device-to-device%20wireless%20communication%3A%20a%20measurement%20driven%20study.%20In%3A%20Wireless%20Communications%20and%20Mobile%20Computing%20Conference%2C%20pp.%20409%E2%80%93414.%20IEEE%20%282017%29)

42. Szóstak, W.: Implementation of a resource-aware method for parallel D2D data streaming. https://bitbucket.org/multiishare/wifi-direct (https://bitbucket.org/multiishare/wifi-direct). Accessed 10 Feb 2021

Copyright information

About this paper

Cite this paper as:

Saganowski S., Kazienko P. (2021) A Resource-Aware Method for Parallel D2D Data Streaming. In: Nguyen N.T., Iliadis L., Maglogiannis I., Trawiński B. (eds) Computational Collective Intelligence. ICCCI 2021. Lecture Notes in Computer Science, vol 12876. Springer, Cham. https://doi.org/10.1007/978-3-030-88081-1_52

- First Online 30 September 2021
- DOI https://doi.org/10.1007/978-3-030-88081-1_52
- Publisher Name Springer, Cham
- Print ISBN 978-3-030-88080-4
- Online ISBN 978-3-030-88081-1
- eBook Packages Computer Science Computer Science (R0)