Implementation Challenges of Single-hop, Device-to-Device Communication in 5G Networks

Marios Sophocleous
Andreas Karkotis
Loizos Christofi
R & D Department
eBOS Technologies Ltd
Nicosia, Cyprus
marioss@ebos.com.cy
karkotis@ebos.com.cy
loizos.christofi@ebos.com.cy

Tomás Lagos Jenschke
Marcelo Dias de Amorim
Serge Fdida
Sorbonne Université and CNRS
Paris, France
tomas.lagos-jenschke@lip6.fr
marcelo.amorim@lip6.fr
serge.fdida@sorbonne-universite.fr

Jose Costa-Requena
Saimanoj Katta
R & D Department
CUMUCORE OY
Espoo, Finland
jose.costa-requena@cumucore.com,
saimanoj.katta@cumucore.com

***Abstract*—We consider the important emerging scenario of a private 5G packet core supporting proximity services enabling coverage extension for end-devices though relays using WiFi-Direct connectivity. We demonstrate and evaluate the implementation of Device-to-Device communications using OTS User Equipment through the development of a customised 5G Packet Core with Local Area Network capability and a mobile/server application to allow the direct communication, discovery and relay selection between the end-device and the relay(s). The system was tested in a laboratory-based test-bed and latency, throughput and jitter measurements were obtained for multiple devices. We conclude that 5G networks are suitable for industrial applications although current 5G solutions are focused in consumer communications, which require changes in the configuration to have similar capacity and delay both for download and upload traffic. Thus, only Non Public Networks (NPN) with proper frame structure and Sub Carrier Space (SCS) can deliver 5G networks better suited for symmetric communications.**

***Keywords*—5G, Device-to-Device Communications, WiFi Direct, 5G LAN, End-device, Single-hop.**

I. Introduction

5G networks have been slowly introduced in our everyday lives, either as Stand-Alone (SA) or as Non-Stand-Alone (NSA), allowing a plethora of benefits to the users such as, improved speeds and lower latency [1]. However, albeit the numerous benefits, 5G technology faces significant challenges, such as coverage range and cost of infrastructure. One approach to address the coverage range and infrastructure cost is the concept of Device-to-device (D2D) communications [2]. Instead of expanding the coverage from each Base Station (BS), User Equipment (UE)s could act as relays enabling single or multi-hop networks, extending the coverage without the need of extra infrastructure [3].

D2D was integrated into 3rd Generation Partnership Project (3GPP) Release 12 to develop a global standard for public safety communications [4], [5]. Within 3GPP Proximity Services (ProSe), D2D was seen as a way to extend conventional cellular services. In Release 13, D2D was consolidated as direct communication between two devices, as one UE was allowed to function as a relay for another UE [6], [7]. The requirements for 5G-related services address D2D in two ways. The first approach uses a direct connection between devices without any network entity in between them. The second approach consists of interposing a relay UE between a UE and the 5G network, where the relay UE can use multiple access schemes, such as 5G radio access technology (RAT), Long Term Evolution (LTE), WiFi and fixed broadband.

Thanks to the characteristic of not requiring a fixed network structure [8], D2D has been the candidate for several applications. An example of this can be seen in using D2D as an alternative content delivery method to address highly demanding and efficiently distributed communications, such as emergency services and natural disasters [9], [10], [11]. Furthermore, we observe a revitalised focus on D2D communications in far-edge architecture, thus opening up more efficient and coherent ways of integrating the overall architecture [12], [13], [14].

There are various open issues and challenges that need attention in securing a D2D communication in 5G environments [15]. Most important challenges in D2D communications are within the context of device discovery (synchronization, initial device discovery signals, multicell device discovery, frequency of discovery messages); in interference management (Cell Densification and Off-loading, D2D in mmWave Communication); in security (Balancing Security-Energy Trade-off, Lack of Standardization, Decentralized Anonymity Schemes, Privacy); in power control (One Large or Multiple Small Networks, Optimal Transmission Power); and in mode selection (Mode Alterations Volume, Mode Selection Overhead, Dynamic Mode Selection). Nevertheless, since infrastructure cost is an important barrier, implementation of D2D needs to be completed with commercially-available UEs. However, such devices do not have all the required features to support this approach.

In this paper and in the framework of the EU-funded project IoT as part of Next Generation Internet (IoT-NGIN)[16], we present some real implementation challenges of D2D communications using common, commercially-available UEs in an attempt to test the end-to-end functionality and performance of single-hop D2D communications. A customised 5G Packet Core (5GC) and a mobile and server

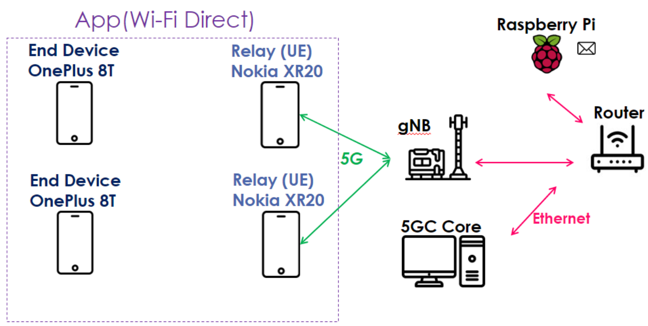

Fig. 1. Overall architecture of the experimental setup.

application have been developed and used in these experiments in order to allow the implementation of a 5G Local Area Network (LAN) so that multiple relays could be identified. The end-device and the relay are connected through Wi-Fi Direct since there is no commercially-available UE that can act as a 5G Time Sensitive Network (TSN) bridge. Finally, the application was also capable to perform a relay selection ensuring power efficiency of the network.

II. Experimental Setup

An end-to-end experimental setup was implemented involving 4 UEs (2xOnePlus 8T & 2xNokia XR20), a laboratory-based BS (Amarisoft Callbox Mini) with a gNodeB (gNB) that is compliant with 3GPP Release 16 having 20 MHz bandwidth, a 5GC (CUMUCORE OY) running on a linux-based PC and a custom-made mobile/server application running on a Raspberry Pi 400. Custom SIM cards and settings were used in the mobile phones in order to properly connect to the BS and the 5GC. Both PLMN-00101 and PLMN-99999 were tested by the custom & programmable SIM cards. The connection between the BS, 5GC and the Raspberry Pi was achieved through Ethernet connections to a 1 Gbps router (MikroTik RouterOS). The gNB and Relay devices were connected with 5G NR, in the Stand-alone mode. The End-Devices and Relay devices were connected with Wi-Fi through the developed mobile/server application.

The overall experimental setup and configuration demonstrating the end-to-end, single-hop, D2D functionality of the test-bed and the multiple relays, multiple end-devices situation is shown in Figure 1. The same configuration was used to test the relay selection capability of the mobile/server application.

III. 5G Packet Core

The Private Networks are known as campus or factory networks but they can avantageously be deployed in other verticals to offer their communications infrastructure. Most of the verticals are currently utilising Wi-Fi but 5G bring new benefits tailored for industrial networks, like coverage thanks to the higher transmission power of 5G base stations both indoors and outdoors. Mobility is another major advantage in 5G. Thus, the private networks or Non Public Networks (NPN) are suitable for Industrial communications that are facing a new era of transformation targeting an increased automation and efficiency. The industrial network require ultra reliability hence they are mostly based on wired fixed LAN. WIFi has been the defacto replacement for some of the wired technologies although exposing some shortcomings. NPNs are being considered a key connectivity solution after latest releases of 5G have defined new features tailored for Industry 4.0 applications. The higher coverage and reliability of NPNs will enable the management of wireless communications in the shop floor without any external interference. With NPN the industrial applications can exchange data locally inside the NPN without external networks in order to guarantee the strong security requirements of industrial scenarios. NPNs by default bring the usage of edge computing with data processing on premise.

The deployment of NPNs follow strong optimization to guarantee indoor coverage based on areas that require high availability connectivity. This is translated into a high customisation where NPN owners have total control over the network deployment and configuration, ensuring high efficiency and QoS. Therefore, NPNs provide very high reliability, thanks to the integration of time synchronisation mechanisms and use of dedicated spectrum. In this paper we focus on 5G feature named 5GLAN that is designed specifically for connecting wireless device with fixed LAN. The 5GLAN is a new feature in 5G where the NPN administrator can go to the network management system and create a 5GLAN group. The group includes the list of the General Public Subscription Identifier (GPSI) or the Subscriber Permanent Identity (SUPI) of all UEs that are supposed to use this 5GLAN group to support private communications. The 5GLAN group can use IP or Ethernet type of communication. In addition the NPN administrator may also indicate any of the following additional information: requested QoS, IPv4 or IPv6 communication, static or dynamic IP address, additional IP services (e.g. DNS, Dynamic DNS, DHCP, IMS, egress to Internet), additional Ethernet services (e.g. multiple IEEE 802.1Q VLANs). After the 5GLAN group is created and the mobile devices part of the group are selected, the 5G network will start managing the traffic between wireless and wired devices as native Ethernet communications.

The User Plane Function (UPF) that handles the user data in the 5G network is configured to route the traffic based on detected MAC addresses. The UPF learns the MAC address(es) connected interface between the 5G core network and the fixed LAN based on the source MAC addresses of the download (DL) traffic received on the LAN interface. The UPF learns the MAC address(es) of UE(s) and devices connected to the 5G network based on the source MAC address contained within the Uplink (UL) traffic received on a data session through the interface with the base station.

The UPF forwards DL unicast traffic (with a known destination address) on a packet data (PDU) Session determined based on the source MAC address(es) used by the UE for the UL traffic. The UPF forwards UL unicast traffic (with a known destination address) on a port (PDU Session) determined based on the source MAC address(es) learned beforehand. The UPF responds on behalf of the UEs part of the 5GLAN group to Address Resolution Protocol (ARP) requests and/or IPv6 Neighbour Solicitation requests based on local cache information for the Ethernet PDUs. The UPF responds to the ARP and / or the IPv6 Neighbour Solicitation Request by providing the MAC address corresponding to the IP address sent in the request. With this mechanism 5G can deliver with the 5GLAN seamless Ethernet communications between fixed LAN and 5G connected devices.

IV. Mobile & Server Application

We showcase the operation of the system through an application running on mobile devices that interact with a server through the 5G Packet Core described in Section III.

A. Mobile Application

We developed an Android application that relies on the Google's Nearby Connections Application Programming Interface (API) to establish D2D links among mobile devices [17]. This API provides tools to perform neighbor discovery before triggering a direct connection between two or more devices. All this without the need for a conventional network infrastructure. Nearby Connections uses Neighbor Discovery Protocol (NDP) with Bluetooth Classic or Bluetooth Low Energy (BLE) to exchange beacons for mutual discovery [18]. In theory, the line-of-sight coverage range of Bluetooth Classic can reach up to $100\,\mathrm{m}$, while that of BLE can reach up to $20\,\mathrm{m}$. The trade-off is that Bluetooth Classic consumes more energy than BLE.

Nearby Connections defines two roles for devices to execute NDP: `advertiser` and `discoverer`. Advertisers regularly throw beacons in the air while discoverers listen to the medium to detect the advertisers around it. Advertisers and discoverers cannot establish connections with devices playing the same role. A device can play either one or both roles, however, it will retain the role with which it has established the connection. Therefore, the choice of a device's mode has thus a direct impact on the formation of the network. Once the devices have discovered each other and established the connections, Nearby Connections uses Bluetooth Classic, Wi-Fi Direct, or Wi-Fi Aware as the link technology for data transfers. The effective link technology will depend on which topological strategy one decides to use. Nearby Connections defines three strategies (depicted in Figure 2), namely `CLUSTER`, `STAR`, and `POINT_TO_POINT` Figure 2:

- The goal of the `CLUSTER` strategy (Figure 2a) is to interconnect as many devices as possible. To do so, it allows each device assigned as a discoverer to connect to more than one advertiser. In addition, this strategy uses Bluetooth Classic for the communication link. In this strategy, contrary to the others, a device can act simultaneously as an advertiser and a discoverer, thanks to Bluetooth's flexibility. However, this flexibility comes with a cost, which is a much lower throughput when compared with the other strategies.
- The `STAR` strategy (Figure 2b) focuses on establishing a star topology using an advertiser as the root and discoverers as leaves. Here, the advertisers can connect to more than one device, whereas discoverers are limited to communicating with a single advertiser. For this purpose, the advertiser acts as an access point that maintains Wi-Fi Direct connections with the leaves. The `STAR` strategy is more rigid than `CLUSTER` but leads to better transfer rates.
- Finally, in the `POINT_TO_POINT` strategy (Figure 2c), a link is established between two and only two devices. In this strategy, advertisers and discoverers have to decide with whom they wish to communicate, as they will not be able to communicate with other devices within range (nor advertise or discover) after establishing a direct link. The advertiser functions as a single-link access point, where a Wi-Fi Aware or Wi-Fi Direct access point will be established if the devices support them. This strategy allows for the highest link performance.

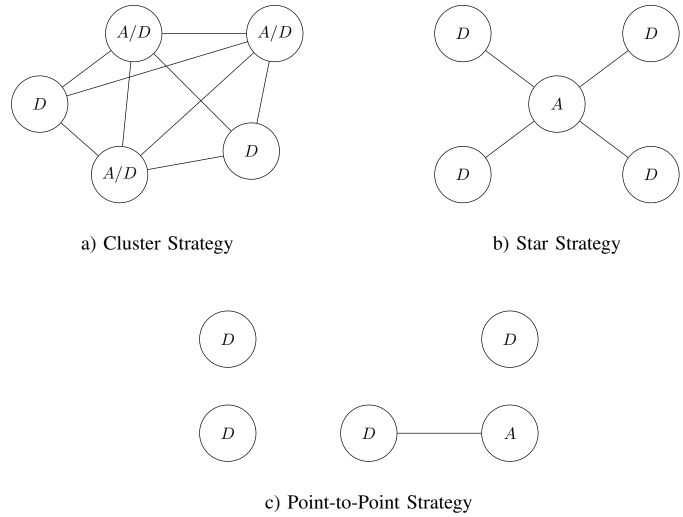

Fig. 2. Nearby Connections's available strategies: (a) `CLUSTER` with two devices set as discover-only (D) and three devices set as advertiser/discover (A/D), (b) `STAR` with one advertiser (which becomes the root) and four discoverers (which become leaves), and (c) `POINT_TO_POINT` with one advertiser and four discoverers (only one discoverer could connect to the advertiser)

Considering the above and based on the requirements within 5G, we adopt the `STAR` strategy in our mobile application. This is because its operation as an access point allows multiple devices to be interconnected and because it fits better the requirements that we set in our work.

B. Server Application

In order to communicate the D2D network with the outside, we have implemented a Transmission Control Protocol (TCP) socket in which the relay devices between the D2D network and the 5GLAN network act as clients, while a Raspberry Pi inside the LAN acts as a server.

When a client connects to the server, the server registers the new client with its unique ID in a registration table. The server then asks the new client for the devices connected via D2D, returning a JSON with the requested list. Likewise, if an already registered relay has updated its table of connected D2D devices, it will notify the server with a new JSON.

Finally, the server generates a routing table where it redirects the traffic of each non-client device to its corresponding Relay, which will then forward the traffic through its D2D link.

V. Results & Discussion

A. Experimental Methodology

Three main tests were performed in this work, the testing of the 5G network and its specifications; the WiFi Direct and its specifications; and the complete, end-to-end communication. The parameters that were used to test the connections between the devices, were:

- Round Trip Time (RTT) is the duration in milliseconds it takes for a network request to go from a starting point to a destination and back again to the starting point. RTT is an important metric in determining the health of a connection on a network, to diagnose the speed and reliability of network connections. RTT is typically measured using a ping. The ping is set to repeats the measurement for 1000 times. Also, the packet size is default, which is 64 bytes [19].
- Throughput is defined as the amount of data per time unit delivered over a physical or logical link or that is passing through a certain network node. It shows the data rate that the network can handle. Throughput depends on factors such as bandwidth, latency, payload size, packet size, network load, number of hops, and others. The measured method is iperf3 [19].
- Jitter is IP Packet Delay Variation. The variation in packet delay is sometimes called jitter. The meaning has to do with the variation of a metric (e.g., delay) with respect to some reference metric (e.g., average delay or minimum delay). The measured method is iperf3 [19].

iperf3 was installed on the server and the other device (client), in order to exchange messages between them. The bandwidth was set to 200 Mbps that is higher than devices can support so that it will allow the devices to reach their maximum speeds. In addition, the reporting intervals is set at 1 s, so every second the system presents bandwidth, jitter and loss reports. Another parameter is the length of the test, which is set at 1000 s. Moreover, for this test it was decided to use User Datagram Protocol (UDP), to primarily establish low-latency and loss-tolerating connection. On the other hand, TCP protocol as a connection-oriented protocol, guarantees the reception of all packets. Therefore, TCP is safer and more reliable than UDP but it is slower and requires more resources. In conclusion, the UDP is better suited for applications that need fast and efficient transmission. The gNB and 5GC operate in a Linux operation where the libraries for these parameters were installed. On the phones, with Android as their operating system, the Android terminal emulator and Linux environment application were installed. This app is a terminal emulation and Linux environment application that works directly with no rooting or setup required.

B. Experimental Results

1) 5G Network: The first set of tests focused on the 5G Network side between the relay devices and the 5GC. The 5GC and Relay device are connected through the gNB using 5G New Radio (NR) in SA mode. Two Nokia XR20 mobile phones were used as relays and they were placed near to the BS during the tests. It must be noted here that these tests were not performed in an anechoic chamber hence, a significant amount of Electro-Magnetic Interference (EMI) was expected. Tables I & II show the obtained results.

TABLE I. Relay Device to 5GC - Download

	RTT (ms)	**Throughput** (Mbps)	**Jitter** (ms)
Phone 1	23.23	129.36	0.15
Phone 2	23.21	96.84	0.19
Average	23.22±0.1	113.10±16.26	0.17±0.02

TABLE II. Relay Device to 5GC - Upload

	RTT (ms)	**Throughput** (Mbps)	**Jitter** (ms)
Phone 1	25.03	3.51	5.53
Phone 2	25.58	1.35	8.05
Average	25.31±0.27	2.43±1.08	6.79±1.26

RTT times, both download & upload, are adequately stable comparing the two relays. In terms of throughput and jitter, it appears that Phone 2 has lower capabilities, both in the download & the upload direction. From this, it becomes obvious that although the devices are the same, there is a significant deviation between their capabilities, something that will eventually affect the overall, end-to-end performance of D2D communication. This highlights again, the erratic behavior of UEs in terms of stability and and huge variation between devices.

A test on the throughput with both PLMN 001-01 and 999-99 was also performed to investigate whether the nature of the network would make a significant difference. The download and upload throughputs for PLMN 001-01 were measured to be approximately 86.8$\pm$27.1 Mbps and 2.52$\pm$1.79 Mbps, respectively. In the case of PLMN 999-99, the download and upload throughputs were measured to be approximately 77.77$\pm$16.27 Mbps and 3.7$\pm$2.33 Mbps, respectively. No significant difference between the two network types was observed, at least not to the level that could be measured with the external noise levels.

2) WiFi Direct Network: The second set of tests is related to the WiFi Direct side of the network. In this test, two OnePlus8T mobile phones were used as end-devices and two Nokia XR20 phones were used as relays. The connection between the end-devices and the relays was achieved through the custom mobile/server application described in the previous section. Tables III & IV show the obtained results.

It is clear from the measurements that this side of the network experiences much higher RTT times, approximately 87.6$\pm$13.49 ms in the download mode. On the other hand, in the upload mode, the RTT times are approximately 21.37$\pm$1.77 ms. Hence, upload RTT times are similar to those of the 5G network but the download side experiences approximately triple RTT times. It is also interesting that for the download RTT, the relay plays the most significant role since both measurements using relay number 1 show higher RTT times. In the upload mode, RTT times are significantly stable.

In terms of throughput, it is obvious that in the download mode, the WiFi Direct side of the network shows speeds

TABLE III. END-DEVICE TO RELAY - DOWNLOAD

	RTT (ms)	Throughput (Mbps)
End Device 1 to Relay 1	101.08	68.55
End Device 1 to Relay 2	75.87	60.97
End Device 2 to Relay 1	93.83	33.29
End Device 2 to Relay 2	79.60	32.89
Average	**87.60±13.49**	**48.93±19.63**

TABLE IV. END-DEVICE TO RELAY - UPLOAD

	RTT (ms)	Throughput (Mbps)
End Device 1 to Relay 1	20.83	25.23
End Device 1 to Relay 2	21.49	45.55
End Device 2 to Relay 1	23.14	48.41
End Device 2 to Relay 2	21.37	41.54
Average	**21.37±1.77**	**41.54±16.31**

of approximately 48.93±19.63 Mbps, which is significantly slower than 5G. It is important to state that in the download mode, the throughput is mostly defined by the end-device and not the relay since both experiments with end-device number 2 show much slower throughput. In this case, end-devices have 2 different Android versions (End-device 1 Build:KB2003 11 C.33, Baseband ver:Q V1 P14, Kernel:4.19.157-perf+; End-Device 2 Build:KB2003 11 C.20 Baseband ver:Q V1 P14, Kernel:4.19.157-perf+), which can significantly affect the network performance. In terms of the upload mode, speeds of 41.54±16.31 Mbps have been recorded showing much higher throughput that 5G.

3) End-to-End D2D Communication Testing: In this last set of tests, the end-to-end setup of this D2D communication was demonstrated. The end-device and the relay connect to each other through WiFi Direct using the custom mobile/server application. The relay is also connected to the 5G BS using 5G NR. Initially, device and relay discovery is achieved through the mobile/server application and secondly message exchange from the server (Raspberry Pi) to the end-device is demonstrated using the configuration shown in Figure 3.

The overall transmission time needed to send the message from the server to the end-device has been measured to be approximately 91 ms with 28 ms standard deviation. Adding the RTT times for the individual segments of travel, the total time is calculated to be approximately 112 ms. Acknowledging that RTT times involve the bi-directional travel of the packet, it is expected that this method would give higher times than

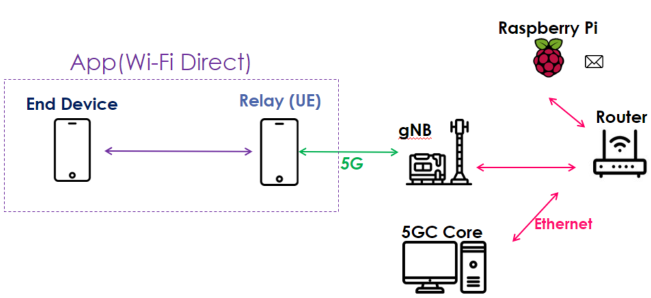

Fig. 3. End-to-End Testing setup

the real ones. However, the calculation and measurements are significantly close to each other confirming the correct order of magnitude in transmission time.

VI. CONCLUSION

In this paper, it has been demonstrated that D2D communications with everyday devices is possible however, there are several remaining implementation challenges in order to use this technology in a seamless environment. It has been demonstrated that although in some cases 5G technology can provide improved performance, offloading the network using D2D with OTS UEs still needs further development of specific technologies that are not yet up to the required level. It has been shown that RTT values of the 5G network are in approximatly 25 ms with throughputs of approximately 140 Mbps download and less than 5 Mbps upload whilst the jitter was less than 0.3 ms download and approximately 30 ms upload. 5G networks are suitable for industrial applications but currently 5G is focused in consumer communications, which require changes in the configuration to have similar capacity and delay both for download and upload traffic. Thus, only Non Public Networks (NPN) with proper frame structure and Sub Carrier Space (SCS) can deliver 5G networks better suited for symmetric communications. WiFi Direct has generally higher RTT times and lower download throughputs compared to the 5G network. Finally, the message transmission from the server application to the end-device has been measured in the order of 100 ms.

ACKNOWLEDGMENT

This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 957246.

REFERENCES

[1] O. Bello and S. Zeadally, "Intelligent device-to-device communication in the internet of things," *IEEE Systems Journal*, vol. 10, no. 3, pp. 1172–1182, 2016.

[2] M. N. Tehrani, M. Uysal, and H. Yanikomeroglu, "Device-to-device communication in 5g cellular networks: challenges, solutions, and future directions," *IEEE Communications Magazine*, vol. 52, no. 5, pp. 86–92, 2014.

[3] U. N. Kar and D. K. Sanyal, "An overview of device-to-device communication in cellular networks," *ICT Express*, vol. 4, no. 4, pp. 203–208, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405959517301467

[4] X. Lin, J. G. Andrews, A. Ghosh, and R. Ratasuk, "An overview of 3gpp device-to-device proximity services," *IEEE Communications Magazine*, vol. 52, 2014.

[5] M. Höyhtyä, O. Apilo, and M. Lasanen, "Review of latest advances in 3gpp standardization: D2d communication in 5g systems and its energy consumption models," *Future Internet*, vol. 10, 2018.

[6] L. Shan, H.-B. Li, R. Miura, and F. Kojima, "Local information sharing system with wireless device-to-device communications," *IEEE Access*, vol. 8, 2020.

[7] F. Rebecchi, M. Dias de Amorim, V. Conan, A. Passarella, R. Bruno, and M. Conti, "Data offloading techniques in cellular networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 17, 2019.

[8] M. Pedhadiya, R. Jha, and H. Bhatt, "Device to device communication: A survey," *Journal of Network and Computer Applications*, vol. 129, 2019.

[9] M. Yin, W. Li, L. Feng, P. Yu, and Q. Qiu, "Emergency communications based on throughput-aware d2d multicasting in 5g public safety networks," *Sensors (Basel)*, vol. 20, 2020.

[10] M. Rihan, M. Selim, C. Xu, and L. Huang, “D2d communication underlaying uav on multiple bands in disaster area: Stochastic geometry analysis,” *IEEE Access*, vol. 7, 2019.

[11] E. Ever, E. Gemikonakli, H. Nguyen, A.-T. Fadi, and A. Yazici, “Performance evaluation of hybrid disaster recovery framework with d2d communications,” *Computer Communications*, vol. 152, 2020.

[12] C. Paniagua, J. Eliasson, and J. Delsing, “Efficient device-to-device service invocation using arrowhead orchestration,” *IEEE Internet of Things Journal*, vol. 7, 2020.

[13] M. H. Adnan and Z. Ahmad Zukarnain, “Device-to-device communication in 5g environment: Issues, solutions, and challenges,” *Symmetry*, vol. 12, 2020.

[14] H. Hussein, M. Radwan, H. Elsayed, and S. Abd El-Kader, “Multi v2x channels resource allocation algorithms for d2d 5g network performance enhancement,” *Vehicular Communications*, vol. 31, 2021.

[15] N. Abedini, S. Tavildar, J. Li, and T. Richardson, “Distributed synchronization for device-to-device communications in an lte network,” *IEEE Transactions on Wireless Communications*, vol. 15, no. 2, pp. 1547–1561, 2016.

[16] I.-N. Consortium, “Home-iot ngin,” 2020, accessed on February 10, 2023. [Online]. Available: https://iot-ngin.eu/

[17] Google, “Nearby Connections – API Overview.” [Online]. Available: https://developers.google.com/nearby/connections/

[18] T. Lagos Jenschke, M. Dias de Amorim, and S. Fdida, “Quantifying direct link establishment delay between android devices,” in *Proceedings of the IEEE Conference on Local Computer Networks (LCN)*, 2022, pp. 214–219.

[19] N. V. Mnisi, O. J. Oyedapo, and A. Kurien, “Active throughput estimation using rtt of differing icmp packet sizes,” in *2008 Third International Conference on Broadband Communications, Information Technology & Biomedical Applications*, 2008, pp. 480–485.