Vehicle Mobility Impact on Performance of Multi-Access Edge Computing

Luiz F. C. Cristino*, Pedro Cruz*, and Luís Henrique M. K. Costa*
* Universidade Federal do Rio de Janeiro - PEE/COPPE/GTA - Rio de Janeiro, RJ, Brazil
Emails: {cristino, cruz, luish}@gta.ufrj.br

***Abstract*—Multi-Access Edge Computing (MEC) is a promising solution that enables limited devices to access external computing resources. This allows users to receive high-performance, low-latency applications while using their User Equipment (UE). MEC relies on servers which are close to the UE and can offer computing resources to UE applications. In a MEC system, low latency is achieved by the proximity of users to the physical resources serving them. However, MEC faces challenges, especially in scenarios where user mobility can disrupt the proximity between users and the physical resources serving them. When the user is moving in a car, this can lead to a challenging environment for applications which trust the MEC to get access to external computing resources with low latency. We investigate this scenario to assess the impact of vehicular mobility on MEC performance. In a realistic setup, we collect nearly 5,000 latency measurements of the path between UEs in a 5G network and evaluate the impact of user mobility on a potential MEC system. We show that distance and signal quality play an important role in the end-to-end latency experienced by the UE.**

***Index Terms*—Multi-Access Edge Computing, 5G, Mobility.**

I. Introduction

The emergence and growing demand for applications that require high levels of computational power is challenging for devices with energy and computational limitations, such as IoT devices, smartphones, or vehicles. As such, limited devices often delegate the processing of complex applications to more robust external computing resources in the cloud. An external computing power handles most intensive tasks, in a process called *offloading* [1]. Some applications may have low latency tolerance for the proper execution of their activities, such as remote robotic surgery and connected autonomous vehicles [2]. In these cases, traditional cloud computing may not be sufficient to provide the computing power under low latency, making edge computing the best alternative. The edge is able to provide external computing resources that are geographically and topologically closer to users [3].

The Multi-Access Edge Computing (MEC) standard developed by the European Telecommunications Standards Institute (ETSI) defines the provision of edge computing resources by Mobile Network Operators (MNOs). In this context, the main devices are mobile and limited, called User Equipment (UE). Each UE can run local applications (UE applications) and, if these applications are too resource-intensive, the UE might need to offload some task. According to the MEC standard, MNOs deploy servers in the edge of their networks, referred to as MEC hosts. MEC hosts are responsible for running MEC Applications (MEC Apps). MEC Apps can handle offloading requests from the UE applications, relieving the UEs from executing intensive tasks. However, there are many challenges in implementing the MEC standard when it comes to dealing with multiple UEs simultaneously, mainly in developing strategies to optimize resource allocation [4]. As a UE moves, it may move away from the MEC host originally allocated to serve it, increasing latency and degrading the QoE. Therefore, resource allocation strategies used by MNOs must consider mobility as a fundamental factor so that UEs are always served by servers that are close enough [5]. Consequently, it is necessary to create and analyze datasets that correlate latency and mobility within the same mobile network in order to improve the corresponding allocation decisions.

This paper investigates the impact of the mobility of a car on the latency between a potential MEC application and a UE application[1]. To do this, we develop two Android applications which instantiate a client and a server within the mobile network of the same MNO, creating a topology similar to that between a UE and a MEC host. Then, we conduct several experiments in which the client sends requests to the server and the server returns a response, while both devices collect metrics about the network and link between them and their respective base stations (BS). In the experiments the client moves around in a car while the server remains static. In addition to the latency values, the collected dataset contains information regarding channel conditions (e.g., signal strength), user equipment location (i.e., Global Positioning System coordinates), connected cell identification (Cell ID), and other values. These metrics allow the estimation of the latency between the two devices and the characterization of the correlation between latency and other aspects, including the distance between the two devices, the distance between the client and its base station, and the signal quality. We also collect the position of the base stations (BSs) from Unwired Labs[2], also aiming to understand the role of the distance between UE and its BS in the latency. The collected data

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) and Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro (FAPERJ).

[1]Some ideas and results of this work are based on a preliminary paper, published in Portuguese in a Brazilian workshop (https://sol.sbc.org.br/index.php/wgrs/article/view/30091). This utilization is permitted by the Brazilian publisher, as seen in https://sol.sbc.org.br/index.php/indice/conduta

[2]https://us1.unwiredlabs.com/

is made available in a public repository[3], and three main contributions can be summarized as follows:

- Production of communication traces between devices within the same MNO;
- A study of the relevance of the distance between MEC host and UE to the communication latency;
- A study of the relevance of the distance between UE and the BS to the communication latency.

A number of papers in the literature address the effect of user mobility to the data traffic. Therefore, mobility traces are obtained that reflect the network conditions while the UE is connected to the mobile network and exchanging information with servers external to the operator [6]–[8]. Additionally, there are studies that theoretically characterize the latency between UEs and the edge [9]. On the other hand, no publicly available 5G traffic traces within the same cellular network have been identified in the literature. Thus, we build in this paper a public dataset capable of emulating traffic between a MEC server and a UE in a 5G network, facilitating the development of efficient allocation and offloading strategies.

This paper is organized as follows. Section II reviews relevant literature. Section III addresses the variability and escalation of latency as a device relocates from proximity to an edge server. The data collection methodology and the data to be gathered are delineated in Section IV. An interpretation of the generated results is presented in the analysis of Section V. Finally, the conclusion is presented in Section V-C.

II. Related Work

Several studies focused on generating datasets to evaluate network capacity in scenarios where UE moves while exchanging packets or requests with an external server. These efforts aim to analyze latency and variations in connection performance in mobile networks or to experiment with different MNO topologies. Raca *et al.* provide a valuable dataset to assess the fluctuation of channel conditions in 5G networks [6]. In their experiments, UEs either download files or stream videos from a cloud server under both stationary and vehicular movement conditions. Latency is measured using a network monitoring application, revealing an average latency increase of 15 ms when the device is in motion. Additionally, the same research group conducted a similar study under 4G conditions, employing the same methodology to produce a dataset [10].

Xiao *et al.*, employing the TCP protocol in 4G networks, introduced a dataset to analyze cellular network performance under high-speed conditions [7]. Their study used UEs located in a high-speed train (up to 300 km/h), a car (up to 100 km/h), and a stationary point. They observed latency as high as 150 ms on the train, compared to around 30 ms in the car and at the stationary point.

Another important contribution is from Safari *et al.*, who generated an open dataset using stationary and distributed mobile nodes [8]. This is done by collecting network data from dozens of MNOs in six countries from the MONROE platform. MONROE is an open source platform that performs measurement campaigns. Nevertheless, their measurements capture interactions between UEs in the MNO network and servers external to the network, rather than within the same network.

In the theoretical domain, several works have analyzed latency between devices and edge servers. Ko *et al.* provide a theoretical analysis of the latency encountered by MEC users, emphasizing the importance of latency assessment for MEC service design [9].

Research on mobility and its effect on network connectivity can also be found outside the context of MEC. Mehmeti and Porta, analyzing a public dataset from a mobile operator in Ireland, show that the time UEs remain connected to the same BS while in motion follows a Pareto distribution [11]. Bouchelaghem *et al.* created a dataset based on user behavior, capturing location traces within a 500-meter range of BSs every five minutes. Their work demonstrates how mobility prediction based on user movement history can enhance MEC systems by optimizing service delivery along user routes [12].

While the aforementioned studies offer valuable insights into the impact of mobility on mobile networks, they primarily focus on the communication between UEs and servers external to the mobile network. None of these studies investigate intra-network communication between devices within the same cellular network, which would enable the evaluation of mobility's effect on potential MEC systems operated entirely within an MNO. Additionally, there is a gap in the literature regarding datasets that capture both mobility and network performance metrics in the same experiment. Our work provides a novel dataset which supports the development of resource allocation strategies designed to optimize performance in the context of UE mobility.

III. Latency Between Device and Edge

To offload a task, an application running on a UE requests an external server to execute the task and return the results. If the server belongs to a cloud service, the server can be located far away from the UE – both geographically and topologically. If the distance between UE and the server is significant, this means that there can be significant communication latency. In this case, even with negligible task processing time, the transmission and response delays can hinder latency-sensitive applications. Using edge computing can reduce this communication delay.

The edge provides computing power closer to the UEs. As the packets have to travel through less hops, the edge reduces the communication latency. The edge also reduces the traffic in the core of the network. One way to implement an edge computing service in the mobile context is through the MEC standard.

A. Multi-Access Edge Computing

According to the MEC standard [13], MNOs distribute offloading-capable servers, named MEC hosts, throughout their access network. A MEC host can be placed in a BSs

[3]https://github.com/GTA-UFRJ/WolfLatency

or at other strategic point in the network. The MEC hosts are managed by the MEC system. The UEs run local applications, called UE applications, that have an interface and interact directly with the end user. The UE applications are able to offload tasks to applications named MEC Apps, which are executed by the MEC hosts. When the UE application decides to offload a task to a MEC host, it first requests the MEC system to instantiate the correspondent MEC App. The MEC system then decides which MEC host should handle the requests from this UE and instantiates the MEC App in the chosen MEC host. The UE application can then proceed to request the MEC application for the the task offloading.

In the MEC standard, the MEC system is responsible for choosing which MEC host should run each MEC application. Choosing the best server is not trivial. The choice depends on the MNO's allocation policy [14] and will only be made if the user and his UE applications have the necessary permissions and no restrictions. In this sense, the MEC system is responsible for the resource allocation, and it should follow some resource allocation policy. Developing an allocation policy is complex and takes into account not only latency due to distance between server and client and their mobility, but also resource availability, network utilization and other factors [5], [15]. In this paper, we focus on the latency and mobility aspects.

B. Latency Between Device and Edge Under Mobility

Figure 1[4] depicts a scenario in which a UE is executing a UE application within a moving vehicle. At instant $t = 1$, the MEC system instantiates the MEC application on the MEC host that is the closest to the UE. Nonetheless, as the UE moves, it moves away from the server that was previously assigned to serve it and towards another server. At instant $t = 2$, the UE is closer to another MEC host. As a result, the optimal server to serve the UE is not the initial MEC host, in term of latency.

In addition to the distance between the server and the UE, the latency between the server and the client also depends on the topology of the MNO's internal network. The topologies of MNO networks can vary significantly and may include not only the access network but also the core network and, in some cases, additional services. In practice, to avoid competitive disadvantage and security breaches, MNOs treat their topologies as proprietary information and do not make it public. Nevertheless, it is not necessary to know the exact internal network topology to measure the latency between two devices. It is sufficient that both are able to exchange information on the same network, so that the round trip time can be obtained.

While ideal, it is not easy in practice to deploy a MEC host in the edge of some MNO and then obtain communication metrics from a UE. The direct access to a BS would be the optimal scenario to avoid interference, wireless communication between devices is sufficient to emulate communication between a user and a MEC server. The internal network of the MNOs is usually protected by firewalls and uses NAT to block incoming requests. Section IV details our data collection procedure to cope with these challenges.

IV. Data Collection Procedure

To collect experimental data, we develop two applications for Android devices [16]. The applications work as a client and a server that exchange requests and responses collecting network data during the procedure. We have named the applications WolfClient and WolfServer. The code of the WolfClient and WolfServer is available in a public repository[5].

We have decided to perform the data collection within the network of one of the three largest MNOs in Brazil, as determined by Anatel's market share ranking of operators [17] (Anatel is the Brazilian National Telecommunications Agency, "Agência Nacional de Telecomunicações").

Our preliminary experiments revealed that, when a device is within the the selected MNO network, it is not possible to reach its IPv4 address, even if the other device is in the same network. This situation was confirmed using the application the ping command and also using HTTP requests. Therefore, if two devices within the selected MNO network try to communicate using IPv4, the traffic between them is blocked for the evaluated protocols. Nevertheless, the preliminary experiments also showed that the IPv6 is reachable and the IPv6 traffic is possible between two devices within the network. As a consequence, we decide to use the IPv6 protocol to collect data from two devices on the same mobile network, overcoming the MNO's barriers.

This paper presents the results obtained using the WolfClient and WolfServer applications, which are responsible for collecting data from the client device and the server device, respectively. The applications are installed on two UEs connected to the same cellular network and exchange Hypertext Transfer Protocol (HTTP) requests [18] with each other over IPv6.

A GET request is initiated by the client, which is running the WolfClient application, and is transmitted to the server, which is running the WolfServer application, using the IPv6 address of the server's UE. The WolfServer receives the request and responds with a basic web page to the WolfClient. Both devices then proceed to store the network conditions experienced by each UE. To infer the latency experienced by the application, we consider the time between an HTTP request is sent by the client and an HTTP response arrives from the server. In addition to this HTTP application latency, we also collect physical layer data using Android libraries. After collecting the metrics and calculating the latency, the client sends a POST to the server containing the collected metrics in one line of information, so that it can also save it as a backup and avoid loss. This POST is not included in the latency calculation to avoid skewing the results. The data generated from these experiments will aid decision making and improve the accuracy of allocation strategies in future work.

[4]Figure with icons from Freepik, from Flaticon.com.

[5] https://github.com/GTA-UFRJ/WolfLatency

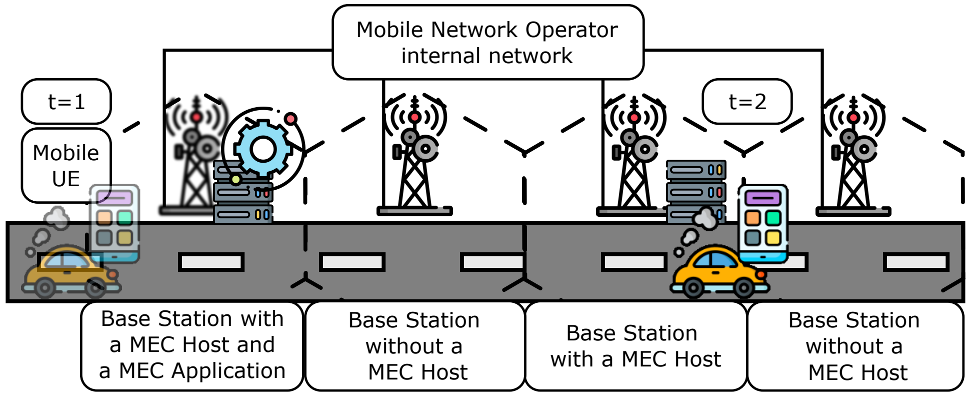

Fig. 1. UE moving in relation to a MEC server.

A. Equipment Used

The experiments employ two UE devices. One device runs WolfServer and is maintained in a fixed position to simulate a potential MEC server. The other device runs WolfClient and is moved through a trajectory, generating data over the network conditions.

TABLE I
DEVICES USED IN THE EXPERIMENTS.

Device name	**Model**	**Function**
UE_1	Galaxy A54 5G SMA546E/DS	Server
UE_2	Galaxy Tab A9+ 5G SM-X216B	Client

Table I provides a summary of the devices utilized in our experiments. Both devices are equipped with 5G connectivity and are configured with SIM cards that are valid for use with data plans.

B. Collected Metrics

The applications WolfClient and WolfServer exchange messages and collect relevant data about the communication in both devices. The data collected is summarized in Table II. They gather information about the transportation means (e.g., walking, train, or car), the date and the time of the measurement (timestamp), the latency experienced by the application (latency), and about the geographical location of the device (latitude and longitude). Additionally, the application gathers data regarding the specific mobile network utilized, including the generation (5G, 4G, 3G, or 2G), the signal strength and level, which are expressed in decibel-milliwatts (dBm), the tracking area code (TAC) and location area code (LAC), the mobile country code (MCC), the mobile network code (MNC), and the cell identification (Cell ID). Furthermore, signal quality data is gathered, including the following parameters: Reference Signal Received Quality (RSRQ), Reference Signal Signal to Noise Ratio (RSSNR), and channel identification (New Radio - Absolute Radio Frequency Channel Number - NRAFCN).

The data collected for identification purposes enables the localization of devices using their latitude and longitude coordinates via the Global Positioning System (GPS). The MCC code is employed to ascertain the country where the device is connected to the Internet, whereas the MNC code is utilized to identify the MNO providing the Internet service. During the course of the experiments, both codes remain constant, as they belong to the same network. Nevertheless, they are retained for comparison with datasets from other regions and other MNOs. The area location codes, TAC/LAC, are employed to delineate the region where the mobile network operator's base station is situated. This allows the base station to be identified by its unique identification number, designated as *cellId* [19].

We use the cellId and the TAC/LAC as input to the Application Programming Interface (API) provided by Unwired Labs. The API returns the GPS coordinates of the BS to which the UE is connected. We attach the coordinates of all the BSs found to our dataset, to enhance future analysis.

The data collected for the purpose of measuring signal quality includes the information on the channel conditions between the UE and its BS. The dataset includes the received signal strength indicator (Signal_dBm), which varies according to the generation of mobile networks to which the UE is connected. The Signal_dBm value corresponds to the Received Signal Strength Indicator (RSSI) when the user is connected to a 2G network, the Received Signal Code Power (RSCP) when connected to a 3G network, and the Reference Signal Received Power (RSRP) when connected to a 4G or 5G network. These signals are quantified in decibels relative to one milliwatt (dBm) and are represented in the signal level indicator for interpretation by the user. The user interface typically displays this indicator in the form of bars ranging from levels 0 to 4. In addition to data regarding the strength of the signal experienced by users, we collect indicators that reflect the quality of the signal received (RSRQ) and the mean value of the signal-to-noise ratio of the cell in question, the Reference Signal Signal to Noise Ratio (RSSNR) [20]. We also collect the new radio absolute radio frequency channel number (NRARFCN), that is available for 5G-NR networks. This measurement is used to identify a specific channel in 5G links. Both the WolfClient and the WolfServer capture signal quality information about the device running them.

TABLE II
METRICS CAPTURED BY THE WOLFCLIENT AND WOLFSERVER.

Field	Definition	Field	Definition
Transportation	Means of transport used	TAC/LAC	Cell area code
Timestamp	Date and time of request	MCC	Country mobile code
Latency	HTTP Request/Response time in ms	MNC	Country operator code
Latitude	GPS Latitude Coordinate	CellId	Identification of the connected cell
Longitude	GPS longitude coordinate	RSRQ	Received signal quality
Mobile_Network	Connected to 2G, 3G, 4G or 5G	RSSNR	Amount of noise in communication
Signal_dbm	Power of the signal obtained, in dBm	NRARFCN	Radio channel number
Signal_level	Signal_dbm translation in levels	-	-

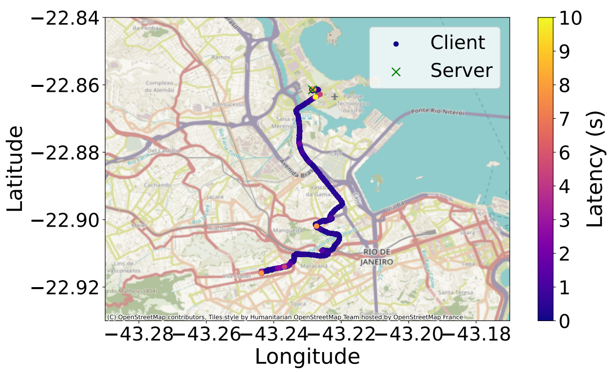

Fig. 2. Route used for the experiments.

C. Data Capture on the Move

We conduct the experiments in the city of Rio de Janeiro, Brazil. In all experiments, the server remains stationary in an open space, while the client moves onboard a vehicle. A map illustrating the positions of the UE and the paths traversed is shown in Figure 2.

Table III presents the number of data points collected in the experiment, the number of cells identified, and the mean values for other variables, including total travel time, distance traveled, and average speed.

In the experiments, device UE_1 hosts the server application, and device UE_2 hosts the client application. The server is located at an open space on the university campus, while the client drives to another neighborhood, in the same region of the city, traveling a distance of to 5.8 km from the server. Figure 2 illustrates the position of the server and the path of the client. The experiment covers both high speed roads and slow areas.

To initiate the experimental procedure, the IPv6 address of the WolfServer device and the mode of transportation employed are entered manually in the WolfClient application. The client initiates one HTTP request per second to the server, except in instances where the server's response exceeds one second. With each request, the client updates its location and network condition information. Similarly, with each received request, the server updates its own dataset with its continuous location and respective network data, thereby establishing a difference in connection status between the two ends of the communication. Furthermore, the data captured by the client is stored in a backup file on the server to prevent the client's data from being lost.

TABLE III
DATASET OVERVIEW.

Attribute	Value
Number of data points	4,825
Traveled distance (mean)	5.7 km
Total time (all routes)	82 min
Average speed	23.61 km/h
Number of cells	96

V. DATASET ANALYSIS

Section IV-C outlined the methodology for capturing network traces under mobility, resulting in the creation of a dataset. In this section, we evaluate the experimental results to understand the relationship between UE proximity to edge resources and communication latency.

A. Relationship between Latency and Distance

Figure 4 shows the latency as a function of distance, with latency presented on a logarithmic scale. The red line represents a least-squares linear fit, which surprisingly shows a negative slope of -51.56 ms/km, indicating a reduction in latency with increasing distance. Nonetheless, the low coefficient of determination ($R^2 = 0.0177$) suggests that distance is not the primary factor affecting latency. Latency varies throughout the route, with most delays clustering within a range which decreases with distance. Nevertheless, there is a significant presence of higher latency, particularly at the beginning of the route, though some outliers persist toward the end. Some delays exceed 1 s, reaching up to 14 s, which could critically impact QoE for latency-sensitive applications like gaming or remote control of equipment. As highlighted in [21], these latency values are far beyond the acceptable threshold of critical applications, typically requiring latency below 10 ms. We explore these unexpected delays in more detail in Section V-C.

The figure 5 highlights that distance between UE and BS is not a reliable predictor of latency in mobile networks. The data points are mostly clustered around specific distances (0-2 km, 4-5 km, and 8 km). In urban settings, the typical distance between a UE device and a MEC server is observed to range between 5 and 10 km. Despite some clusters of lower latency

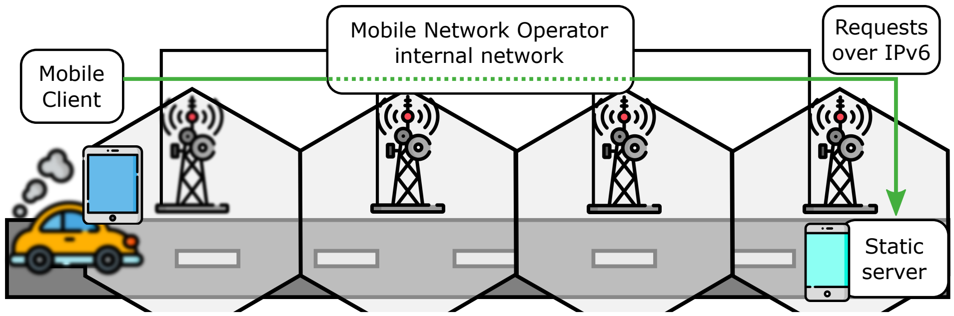

Fig. 3. Scenario of the experiments.

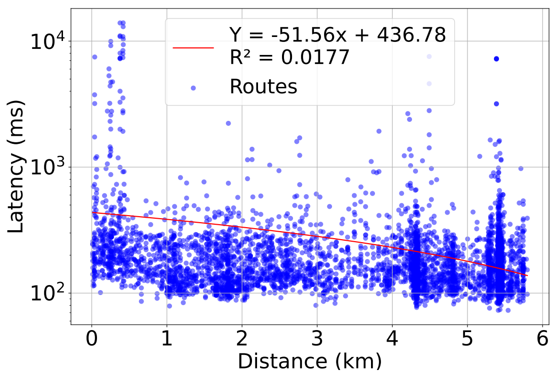

Fig. 4. Latency as a function of the distance between server and client.

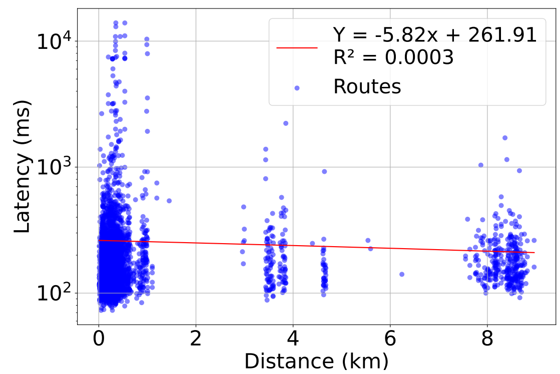

Fig. 5. Latency as a function of the distance between base stations and client.

at short distances, the variability and presence of significant outliers suggest that other factors, such as network congestion, signal quality, multipath, or infrastructure issues, are more influential in determining latency than the simple physical separation between base stations and clients. The near-zero value ($R^2 = 0.0003$) supports this hypothesis.

Moreover, latency spikes at shorter distances may be attributed to local network conditions, including interference, high user density, or transmission errors that require packet retransmissions, increasing delays. The observed lack of a strong trend in the relationship between distance and latency reinforces the notion that signal quality (influenced by factors like obstructions, radio frequency interference, or weather conditions) plays a much larger role than proximity alone.

The jitter (calculated according to RFC4689 [22]) is shown in Figure 6. The figure illustrates how jitter varies with distance, revealing the effects of mobility on packet transmission. High jitter can severely impact real-time services, such as voice communications and video streaming, by causing delays and dropped frames. The figure suggests that while there is a slight downward trend in jitter with increasing distance, distance alone does not significantly influence jitter. The high level of variability indicates that jitter is likely driven by a complex set of factors, including network conditions, interference, and environmental dynamics, which are not captured by distance alone. It was expected that other factors would affect jitter, but not that distance would have such a small effect.

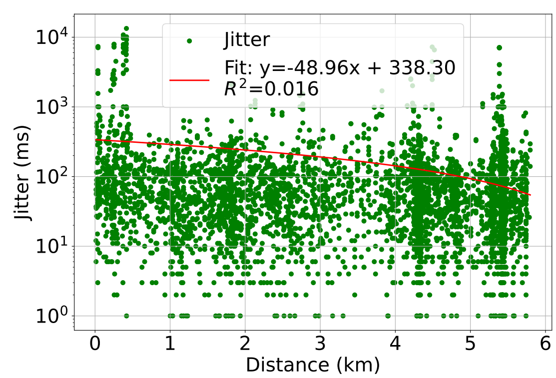

Fig. 6. Jitter experienced along the routes.

Figure 7 shows the route taken by the vehicle and the locations of the BSs the client connected to during the experiments. A total of 96 cells were observed across the route, with the UE alternating between 4G and 5G technologies. Notably, the 5G connections were based on 4G infrastructure, utilizing Non-Standalone (NR-NSA) technology [23]. This configuration causes the UE to recognize the network as 4G with 5G data support. Of the 96 cells, 82 supported 5G connections, while 55 supported 4G, reflecting the current state of 5G deployment of the MNO, which uses dual technology capabilities.

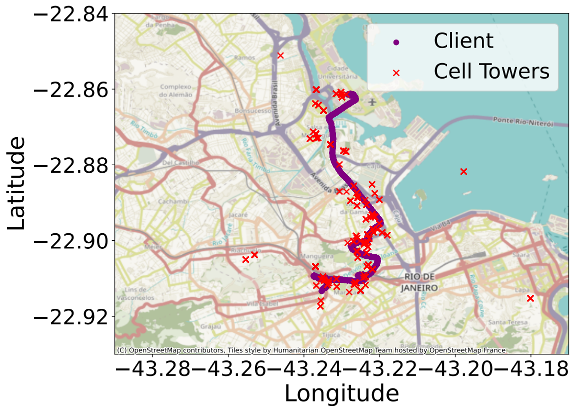

Fig. 7. Cell Towers and Client Positions obtained from all routes.

Latency maps in Figure 8 depict the latency experienced during the experiments. Higher latency areas are represented by yellow, while lower latency regions are in blue. Higher latency tend to cluster at the beginning of the route. This is attributable to the high concentration of low-level signal data in this area, oscillating between level 1 and level 0. Additionally, there is a considerable amount of noise interference (RSSNR) with a multitude of negative values, and the received signal quality (RSRQ) remains in the -20 to -15 range and exhibits several changes from 5G to 4G. These values collectively contribute to an unfavorable connection experience.

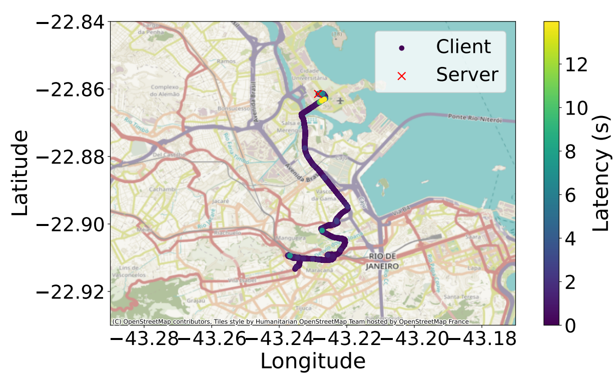

Fig. 8. Latency maps for the different experiments.

B. Evaluation of Minimum Latency per Window

Figure 9 presents the empirical cumulative distribution function (ECDF) of the latency values. In each experiment, an inflection region is observed, between 300 and 600 milliseconds. Notably, Route 1 consistently shows higher latencies, while Route 3 is the first to reach higher latency outliers. A potential explanation is that the latencies stem from both transmission delays and interference, as well as retransmissions due to packet loss.

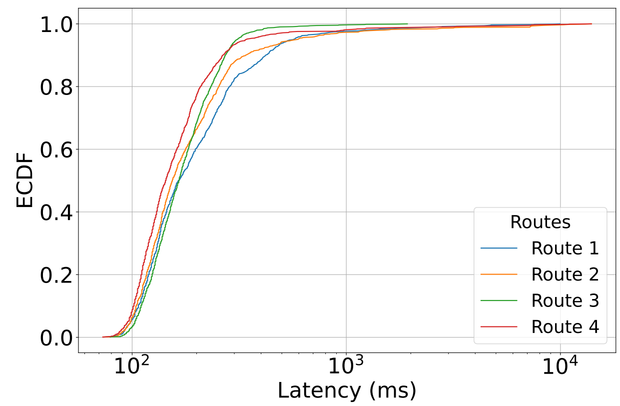

Fig. 9. Cumulative probability function of latency across the routes.

C. Influence of Link Conditions

We also evaluated the effect of signal strength over the latency measured. Figure 10 illustrates the distribution of signal levels, with the majority of the experiment occurring under strong signal conditions. Figure 11 shows a box plot of latency versus signal level, on a logarithmic scale. As expected, lower signal strength corresponds to higher latency, with extreme deviations, such as latencies over 1000 ms, observed when the signal falls below -100 dB.

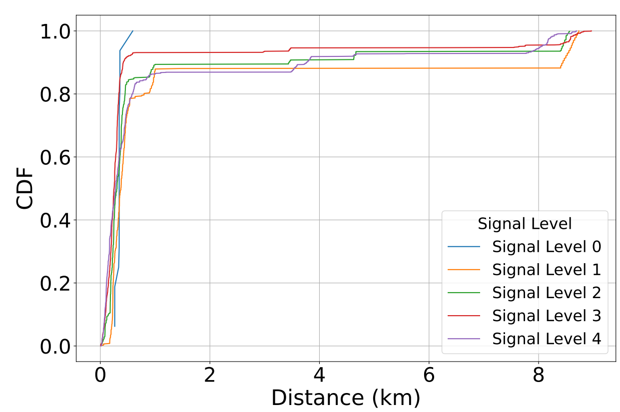

Fig. 10. Proportion of signal levels vs. UE-BS distance

The signal strength significantly influences latency, as shown in Figure 11. Poor signal conditions lead to a wider latency distribution, suggesting that retransmissions and packet loss under poor connectivity contribute to elevated latency. In weaker signal environments, the lower signal-to-noise ratio (SNR) can cause frequent errors in transmission, necessitating the retransmission of packets. This, in turn, increases the latency and introduces jitter.

CONCLUSION

Multi-Access Edge Computing (MEC) is a standard developed by the European Telecommunications Standards Institute (ETSI) that enables Mobile Network Operators (MNOs) to

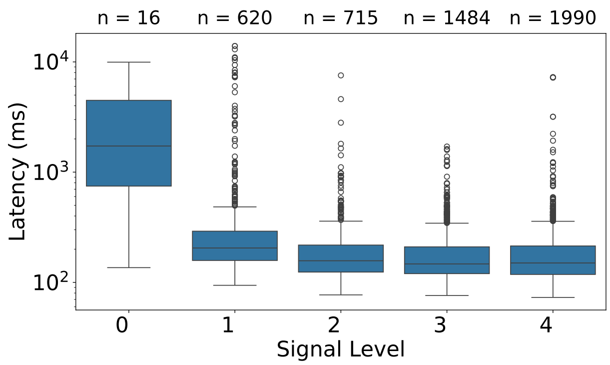

Fig. 11. Relationship between the proportion and latency generated by signal level.

provide edge computing resources to their users. MEC provides the computing power of MEC servers (MEC hosts) to offload tasks from User Equipment (UEs). Furthermore, MEC relies on the proximity between UEs and MEC servers to ensure low latency. However, the mobility of UEs can impede the potential of MEC to provide low latency, as a moving UE moves further away from the MEC server. This effect can be accentuated by the mobility of vehicles.

The objective of this work is to assess the relationship between distance and delay experienced by the UE and the MEC server in the context of vehicle mobility and build a dataset which can be used by other studies. To this end, experimental data was collected from two moving UEs. One UE was designated as the server, while the other acted as the client. Data was gathered from the client's movement, with nearly 5,000 samples collected. In addition to latency samples, signal quality data was also collected. The results are presented on the relationship between distance and latency, as well as the influence of signal strength on latency.

As future work, the dataset generated from the mobility traces is expected to be used in simulation projects, with the objective of creating allocation strategies based on the mobile behavior of UEs in a real network. Thus, new studies should be initiated with the aim of investigating the impact of mobility in MEC system scenarios, emulating the expected behavior and the optimal strategies for dealing with multiple users.

REFERENCES

[1] A. Islam, A. Debnath, M. Ghose, and S. Chakraborty, "A survey on task offloading in multi-access edge computing," *Journal of Systems Architecture*, vol. 118, p. 102225, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1383762121001570

[2] M. A. Lema, A. Laya, T. Mahmoodi, M. Cuevas, J. Sachs, J. Markendahl, and M. Dohler, "Business case and technology analysis for 5G low latency applications," *IEEE Access*, vol. 5, pp. 5917–5935, 2017.

[3] J. Ren, G. Yu, Y. He, and G. Y. Li, "Collaborative cloud and edge computing for latency minimization," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 5031–5044, 2019.

[4] M. Chen and Y. Hao, "Task offloading for mobile edge computing in software defined ultra-dense network," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 3, pp. 587–597, 2018.

[5] Q. Luo, S. Hu, C. Li, G. Li, and W. Shi, "Resource scheduling in edge computing: A survey," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 4, pp. 2131–2165, 2021.

[6] D. Raca, D. Leahy, C. J. Sreenan, and J. J. Quinlan, "Beyond throughput, the next generation: a 5G dataset with channel and context metrics," in *Proceedings of the 11th ACM Multimedia Systems Conference*, ser. MMSys '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 303–308. [Online]. Available: https://doi.org/10.1145/3339825.3394938

[7] Q. Xiao, K. Xu, D. Wang, L. Li, and Y. Zhong, "TCP performance over mobile networks in high-speed mobility scenarios," in *IEEE 22nd International Conference on Network Protocols*, 2014, pp. 281–286.

[8] A. Safari Khatouni, M. Trevisan, D. Giordano, M. Rajiullah, S. Alfredsson, A. Brunstrom, C. Midoglu, and Ö. Alay, "An open dataset of operational mobile networks," in *Proceedings of the 18th ACM Symposium on Mobility Management and Wireless Access*, 2020, pp. 83–90.

[9] S.-W. Ko, K. Han, and K. Huang, "Wireless networks for mobile edge computing: Spatial modeling and latency analysis," *IEEE Transactions on Wireless Communications*, vol. 17, no. 8, pp. 5225–5240, 2018.

[10] D. Raca, J. J. Quinlan, A. H. Zahran, and C. J. Sreenan, "Beyond throughput: A 4G LTE dataset with channel and context metrics," in *Proceedings of the 9th ACM multimedia systems conference*, 2018, pp. 460–465.

[11] F. Mehmeti and T. F. L. Porta, "Analyzing a 5G dataset and modeling metrics of interest," in *2021 17th International Conference on Mobility, Sensing and Networking (MSN)*, 2021, pp. 81–88.

[12] S. Bouchelaghem, H. Boudjelaba, M. Omar, and M. Amad, "User mobility dataset for 5G networks based on GPS geolocation," in *2022 IEEE 27th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, 2022, pp. 128–133.

[13] ETSI, "Multi-access edge computing (mec); framework and reference architecture," European Telecommunications Standards Institute, France, Tech. Rep. ETSI GS MEC 003 V3.1.1, March 2022.

[14] H. Djigal, J. Xu, L. Liu, and Y. Zhang, "Machine and deep learning for resource allocation in multi-access edge computing: A survey," *IEEE Communications Surveys & Tutorials*, 2022.

[15] P. Cruz, N. Achir, and A. C. Viana, "On the edge of the deployment: A survey on multi-access edge computing," *ACM Comput. Surv.*, vol. 55, no. 5, dec 2022. [Online]. Available: https://doi.org/10.1145/3529758

[16] Google, "Android API reference," https://developer.android.com/reference, 2024, acessada em: 28 de Janeiro de 2024.

[17] Anatel, "Ranking das operadoras da agência nacional de telecomunicações," https://informacoes.anatel.gov.br/paineis/acessos/ranking, 2021, acessada em: 28 de Janeiro de 2024.

[18] R. T. Fielding, M. Nottingham, and J. Reschke, "HTTP semantics," *Internet Engineering Task Force, Request for Comments RFC*, vol. 9110, 2022.

[19] S. Han, T. Kang, and J. Seo, "Smartphone application to estimate distances from LTE base stations based on received signal strength measurements," in *2019 34th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC)*, 2019, pp. 1–3.

[20] O. Simpson and Y. Sun, "LTE RSRP, RSRQ, RSSNR and local topography profile data for RF propagation planning and network optimization in an urban propagation environment," *Data in Brief*, vol. 21, pp. 1724–1737, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352340918309892

[21] A. Hassebo, M. Obaidat, and M. A. Ali, "Commercial 4G LTE cellular networks for supporting emerging iot applications," in *2018 Advances in Science and Engineering Technology International Conferences (ASET)*, 2018, pp. 1–6.

[22] S. Poretsky, S. Erramilli, J. Perser, and S. Khurana, "Terminology for Benchmarking Network-layer Traffic Control Mechanisms," RFC 4689, Oct. 2006. [Online]. Available: https://www.rfc-editor.org/info/rfc4689

[23] A. Narayanan, X. Zhang, R. Zhu, A. Hassan, S. Jin, X. Zhu, X. Zhang, D. Rybkin, Z. Yang, Z. M. Mao, F. Qian, and Z.-L. Zhang, "A variegated look at 5G in the wild: performance, power, and qoe implications," in *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*, ser. SIGCOMM '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 610–625. [Online]. Available: https://doi.org/10.1145/3452296.3472923