Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 6651402, 12 pages
https://doi.org/10.1155/2021/6651402

Research Article

MASEMUL: A Simulation Tool for Movement-Aware MANET Scheduling Strategies for Multimedia Communications

Moustafa M. Nasralla,[1] **Iván García-Magariño**,[2] **and Jaime Lloret**[3]

[1]*Department of Communications and Networks Engineering, Prince Sultan University, Riyadh 11586, Saudi Arabia*
[2]*Department of Software Engineering and Artificial Intelligence, Complutense University of Madrid, 28040 Madrid, Spain*
[3]*Integrated Management Coastal Research Institute, Universitat Politécnica de Valéncia, Valéncia 46022, Spain*

Correspondence should be addressed to Jaime Lloret; jlloret@dcom.upv.es

Received 11 October 2020; Revised 2 March 2021; Accepted 7 March 2021; Published 18 March 2021

Academic Editor: Claudia Campolo

The last decade has witnessed a steep growth in multimedia traffic due to real-time content delivery such as in online games and video conferencing. In some contexts, MANETs play a key role in the hyperconnectivity of everything in multimedia services. In this context, this work proposes a new scheduling approach based on context-aware mobile nodes for their connectivity. The contribution relies on reporting not only the locations of devices in the network but also their movement identified by sensors. In order to illustrate this approach, we have developed a novel agent-based simulator called MASEMUL for illustrating the proposed approach. The results show that a movement-aware scheduling strategy defined with the proposed approach has decreased the ratio of channel interruptions over another common strategy in mobile networks.

1. Introduction

Mobile ad hoc networks (MANETs) are composed of flexible nodes that usually join or leave. In MANETs, many optimizations have been addressed to provide high quality of service (QoS) [1]. In addition, QoS-driven approaches have performed data mining for identifying key aspects for improving the MANET performance such as selecting which services to deploy [2]. The ways of improving QoS in MANETs are diverse, and each of these brings a new insight aspect for improving MANETs. Among all these possible aspects, this work focuses on the analysis of device movement in MANETs and the content-aware techniques by specially considering the protocols for transmitting multimedia contents.

The literature has analyzed the movement of devices in MANETs. For instance, smart movement of unmanned air vehicles (UAVs) has been proposed for improving connectivity in MANETs of UAVs [3]. In this context, triangulation heuristics were proposed to avoid the computational cost of solving the corresponding NP-hard optimization problem. The connectivity in high-speed MANETs becomes crucial, and Chen et al. [4] proposed to use link stability probability estimation between nodes to rapidly adapt to the MANET topology among other parameters, including an interruption prediction mechanism. However, these works did not provide a MANET simulator in which alterations of their proposals could be tested to improve the state of the art in this context.

Content-aware strategies can adapt the scheduling of communications considering the type of content transmitted [5]. More concretely, some works focus on specific contents, being particularly common targeting multimedia transmission types as these are the neediest in terms of bandwidth and connection stability. Among others, one can observe the work of Venkatesh et al. [6] about a QoS-aware routing protocol for video streaming in MANETs. In addition, another QoS-aware strategy is balanced between real-time and non-real-time traffic [7]. In general, online interactive transmission may benefit from real-time communication without disruptions being the latter one of the most critical aspects.

In the intersection of movement analysis on MANETs and specific interactive multimedia transmission, this work proposes a novel simulator for supporting further analysis of MANETs composed of devices that include sensors for identifying locations and movements. The proposed simulator called MASEMUL allows scheduling strategies to be aware of locations, movement directions, and speed of the devices conforming the MANET. MASEMUL is open source. For illustrating this approach, we propose a strategy that estimates the nodes that are more stable, in terms of relative movement with the previous and next nodes in the channel, to establish a high bandwidth channel for transferring real-time multimedia content.

Scheduling network strategies usually ignore the movement information when handling networks as one can observe, for example, in [8], which even considered real-time traffic changes. Moreover, even in MANET scheduling works such as [9], the literature lacks support for the definition of movement-aware strategies based on relative movement among network devices. Furthermore, well-known general-purpose network simulators like OMNeT++ and NS-3 also omit the possibility of defining movement-aware strategies based on the movement of the current device and surrounding ones. The main advantage of MASEMUL over other network simulators is its capacity of designing scheduling strategies in a distributed way considering the movement of each device and other surrounding devices. MASEMUL supports the simulation of UAVs among other vehicles.

This article is structured as follows. The next section introduces the related work indicating the main gap of the literature covered by this work. Section 3 presents a novel approach for supporting multimedia real-time communications in MANETs. Section 4 presents MASEMUL, and Section 5 discusses the most relevant aspects of the experimentation. Section 6 mentions the main conclusions and depicts some future research lines.

2. Related Work

The related work falls into the categories of (a) existing scheduling algorithms and approaches, (b) network simulators in the context of this work, and (c) a further comparison with OMNeT++ and NS-3 simulators.

2.1. Network Scheduling Algorithms. Coordinated scheduling algorithms have been used to mitigate intercell interference in different types of networks. In particular, Nardini et al. [8] solved the coordination of cluster of nodes as an optimization problem. Their algorithms obtained suboptimal solutions fast enough to adapt to traffic changes in real time. Their results showed a reduction of used resource blocks, and consequently, their approach reduced energy consumption. In addition, Song et al. [10] improved scheduling by a free space optical framework. They used a relay selection algorithm to adapt to different weather conditions for boosting transmission quality.

Concurrent scheduling has been applied in improving massive multiple input, multiple output (MIMO). For instance, Ma et al. [11] investigated scheduling between backhaul and access link for maximizing QoS-satisfied transmission flows. Their experimentation showed improvements in terms of network throughput and number of successful transmission flows in comparison to other existing schemes.

In MANETs, scheduling also plays a relevant role. In particular, Tuan et al. [9] proposed a channel-aware scheduling mechanism that considers integrated time in frequency-based downlink LTE. They improved around 12% over other methods such as modified largest weighted delay first (M-LWDF), proportional fair (PF), and exponential/proportional fair (EXP/PF).

Video transmission has been improved in different kinds of networks. For example, Sharma et al. [12] proposed a catalytic computing approach for improving resource-based mobility management in video transmission. Their approach relied on a homogeneous discrete Markov model for determining user mobility patterns and an algorithm for congestion prediction and selection of fast routes among serving terminals.

Moreover, the cooperated blockchain networks have been applied to a wide range of use-case domains such as smart healthcare, smart city, smart transportation, smart grids, and UAVs. The survey of Nguyen et al. [13] showed the security challenges in networks that can be solved with blockchain in these scenarios.

The aforementioned related works propose the mechanism of scheduling network packets in different kinds of scenarios. A few of these actually consider MANETs. However, these strategies were not integrated into a simulation environment, where their approaches could be easily simulated for MANETs. Thus, practitioners may find it difficult to test and improve their approaches, missing a MANET simulator for that purpose.

2.2. General-Purpose Network Simulators. The Discrete EVent Simulation framework for MANETs (DEVS-M) was proposed as an adaptation of the previous DEVS-Suite simulator focusing on MANETs [14]. This simulator also checks the velocity and movement of nodes. They tested the load-balancing scheme for illustrating their approach. Their simulator programmed in Java and their load-balanced technique showed that the simulation execution time did not increase with the network size, which was an advantage over other simulators and load-balancing strategies.

The estimation of QoS and also quality of experience (QoE) is a problem widely addressed in the literature. For instance, neural networks have been used for addressing this problem with machine learning using a device-aware and content-aware approach [15]. In particular, their approach used a multilayer perceptron and measured the medical QoE based on the visual perception of medical experts as input and estimated output.

Nevertheless, these simulators are not adapted for MANETs except for the case of DEVS-M, which is basically focused and illustrated on load balancing rather than on the reduction of interruptions in channels of communications with moving devices, which reduces the impact on the quality of transmitted multimedia flows. In this context, this work covers this gap of the literature by presenting a MANET simulator based on easily configurable strategies with distance-based heuristics as described in the next section.

2.3. OMNeT++, NS-3, and Other Simulators. Among all the network framework simulators, OMNeT++ and NS-3 are especially relevant for their popularity and their extensive functionality. Both are designed for supporting discrete event network simulators. OMNeT++ is built as a C++ modular library, on which network simulators can be built. However, it lacks its own simulator. NS-3 is a free network simulator software that supports IP and non-IP-based networks. It supports models such as Wi-Fi, WiMAX, or LTE. Although there are mobility models for OMNeT++ and NS-3, these platforms still lack the support of the direct definition of strategies based on information such as relative speeds.

Table 1 compares the most relevant features of OMNeT++ and NS-3 with the novel proposed MANET simulator named MASEMUL, in the context of this work. These simulators support the simulation of MANETs. OMNeT++ requires using a domain-specific language, while in NS-3 and MASEMUL, it is not necessary to learn a domain-specific language. The comparison also includes the 5G-air-simulator [16] and Yet Another MANET Simulator (http://www.pages.drexel.edu/sf69/MANET_Simulator.html (last accessed on 01/25/2021)) from the Drexel Network Modeling Laboratory. It is worth highlighting that 5G-air-simulator is an extension of the LTE-Sim simulator of Piro et al. [17] from the lab, which was initially developed in 2010. 5G-air-simulator used the same environment with new capabilities and consequently is the result of the work for more than a decade from TelematicsLab.

The advantages of the presented MASEMUL over these existing alternatives are mainly related to the definition of strategies based on the movement properties. More concretely, in MASEMUL, strategies can access the movement information of each device represented as a vector. These strategies can be designed for being specifically aware of movement in an easy and straightforward way. Furthermore, the agent-based simulation approach of MASEMUL guides designers in specifying their strategy in a distributed way, defining the behaviors of individual nodes, and making the network more robust against failures and dynamic circumstances.

3. The Proposed MANET Agent-Based Simulator for Evaluating Communications with Multimedia

The proposed simulator referred to as MASEMUL (MANET Agent-based Simulator for Evaluating communications with MULtimedia) is introduced considering the most relevant underlying basis in Subsection 3.1, the functionalities associated with its architecture in Subsection 3.2, and finally the movement-aware strategies in Section 3.3.

3.1. Relevant Underlying Basis of MASEMUL. In MASEMUL, we have modeled devices as context-aware components that use location and movement information from sensors for scheduling and routing purposes in mobile networks. More concretely, location is represented with a 3D vector. The movement is also represented with a 3D vector with the direction of the movement and the magnitude representing the actual speed. The simulator uses 3D vectors for the locations and movements of vehicles to support vehicles with different altitudes such as UAVs. In order to simplify the visualization, the simulator uses a vertical projection representation, omitting the third z dimension.

MASEMUL considers the types of compression for videos and images. It simulates the time of compressing and decompressing at each end, and it considers the size of transmitted videos considering the corresponding compression factor. Therefore, the QoS metrics reflect the influence of the specific multimedia compression.

The proposed simulator considers devices that are aware of their location and movement (i.e., direction and speed). Each device operates autonomously but coordinates with shared behaviors to keep MANET aware of their information. In particular, we have developed this simulator following the process of developing efficient agent-based simulators (PEABS) [18]. In this context, each device is modeled as an agent, with a shared goal of improving QoS in transmission of multimedia content in MANETs.

3.2. Description of the Functionalities and Architecture of MASEMUL. Figure 1 presents the block diagram of the main functionalities of MASEMUL. In this simulator, devices are created in the beginning with the proper sensors to be context-aware of their locations, directions, and speeds. Their locations, directions, and speeds are randomly initialized. Then, source and target devices are selected for performing the multimedia streaming/conferencing. The simulator selects the scheduling and routing strategy. It establishes the channel according to this strategy. The simulation continues with a loop that simulates the streaming of multimedia content. Each iteration simulates the transmission of a packet and the movement of MANET devices according to their locations, directions, and speeds. Each iteration checks whether the channel is broken because the distance of any two consecutive nodes of the channel may surpass the corresponding threshold. If the channel is broken, then a new channel is reestablished according to the strategy. The simulation updates several QoS measurements. More concretely, it calculates the average communication delays, computed as end-to-end delays. It also calculates the packet delivery ratio and the packet loss ratio. Besides the common QoS measurements, this simulator also simulates the number of times that the channel is broken, as these may cause pauses in real-time multimedia transmissions.

Moreover, MASEMUL has the possibility of simulating several scheduling strategies in the same scenario for performance comparison. It can use the two strategies described in this article or any other implemented by the user. In these cases, everything is repeated for each strategy at the beginning and in each simulation iteration.

The movement-aware scheduling approach uses a greedy algorithm, in which each node selects the next node, and then, it takes the next decision with the available options. This greedy algorithm checks that the channel does not go through the same nodes to avoid infinite loops in the search. In this approach, it is essential to select proper heuristics to facilitate reaching the target device properly.

Table 1: Comparison of OMNeT++, NS-3, 5G-air-simulator, Yet Another MANET Simulator, and MASEMUL.

	OMNeT++	NS-3	5G-air-simulator	Yet Another MANET Simulator	MASEMUL
Can it simulate MANETs?	✓	✓	✓	✓	✓
Can users perform simulations without learning any domain-specific language?	✗	✓	✓	✓	✓
Is it written in a general-purpose programming language (e.g., C++/Java)?	✓	✓	✓	✓	✓
Can you define strategies that directly access the movement vector of each device?	✗	✗	✗	✗	✓
Can it simulate movement-aware strategies?	✗	✗	✗	✗	✓
Can the strategies be defined with an agent-based simulation approach by defining autonomous behaviors in the network nodes?	✗	✗	✗	✗	✓

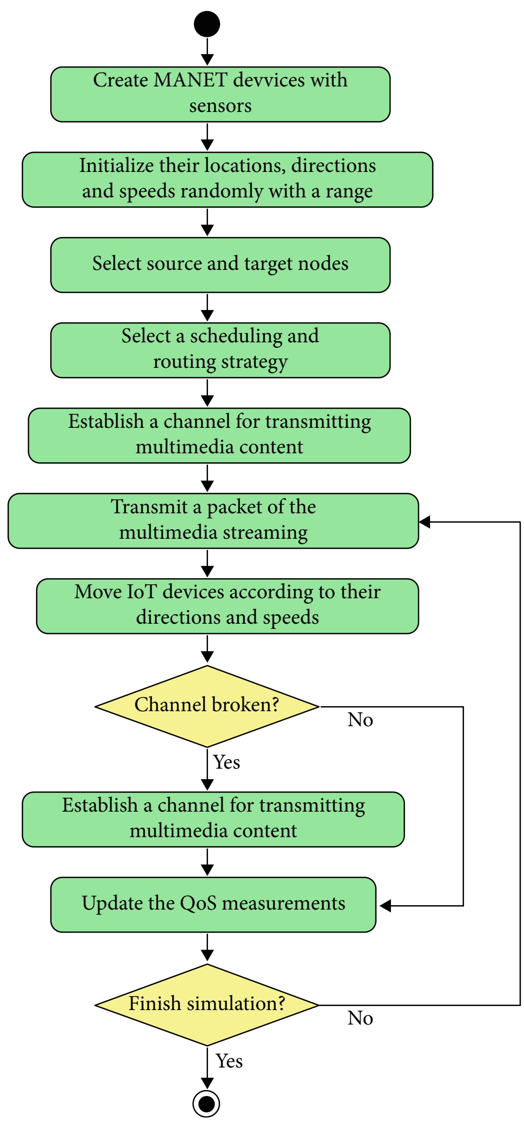


Figure 1: Block diagram of the proposed MASEMUL simulator.

In particular, this framework has an object-oriented "Strategy" interface with a distance method that calculates the heuristic for each node, knowing the current node, the node that is being evaluated, and the target node. Figure 2 shows a relevant excerpt of the class diagram of the MASEMUL. The "MainSim" class guides all the other classes in the simulation. The "Channel" class simulates each communication and can use different movement-aware strategies. Practitioners can define new movement-aware strategies by implementing the "Strategy" interface. Each of these strategies can access the environment and access all the reachable network elements or devices (i.e., objects of "Device" class) and use their movement and relative speed among other information. The "QoS" class monitors the simulation to calculate the QoS measurements.

Practitioners can just test new strategies by just implementing this Strategy interface with its method, using the MASEMUL framework publicly available on its website (http://grasia.fdi.ucm.es/ivan/masemul/ (last accessed 02/01/2021)). This framework includes two examples of strategies that will be presented later in Section 4 when presenting the experimentation. The class diagram shows how the environment contains several devices, and each channel includes a strategy for establishing communications between the target and source devices. Another class measures the QoS indicators, and the environment is graphically displayed with another class.

This approach avoids exponential complexity solutions such as brute-force approaches and backtracking algorithms, which address NP-hard problems with exponential complexity cost. In usual medium and big MANETs, this cost is not affordable for the necessary time that would be needed to calculate paths.

3.3. Simulating Movement-Aware Strategies. Figure 3 shows an example of a simulated network scenario by using the novel MASEMUL simulator. In the simulator, you can establish the simulated time in seconds, the number of devices, the environment size (expressing width of a square area in meters), the maximum distance to communicate among two different devices, the maximum movement speed in each axis, and the simulation display speed (i.e., number of ms for showing a simulated second).

In MASEMUL, the nodes of MANETs are initialized with random locations and movements. Certain strategies can be

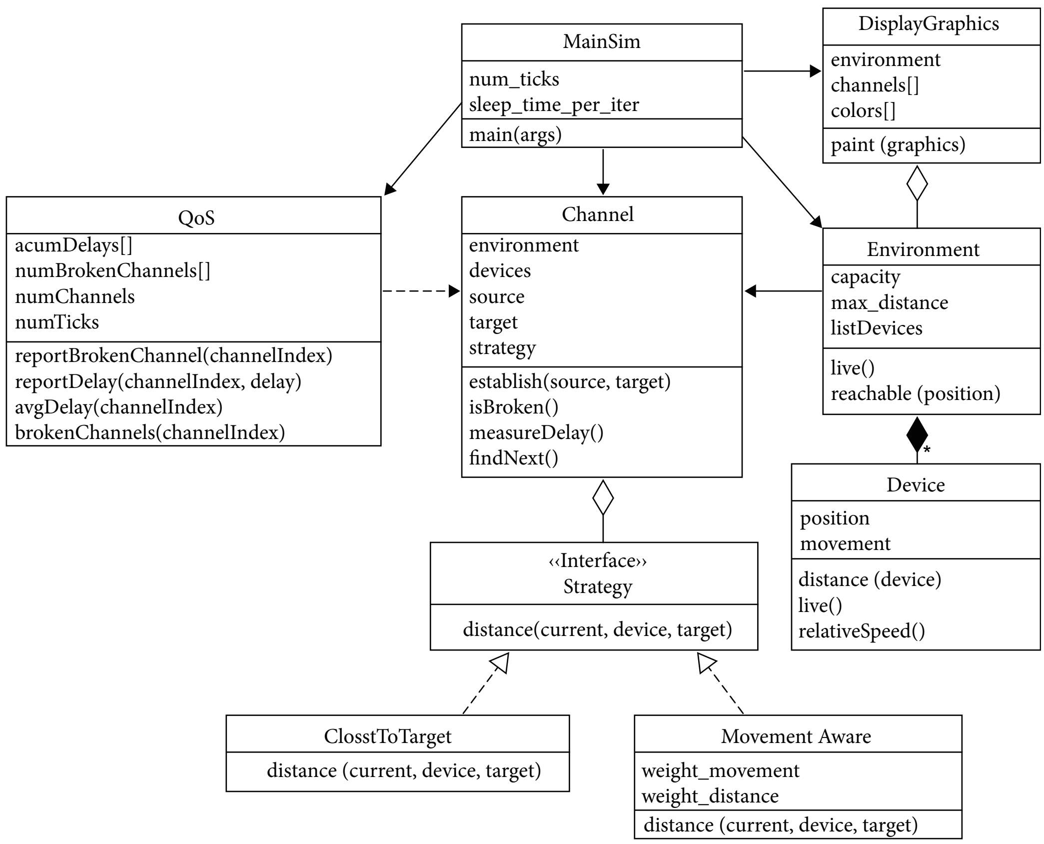


Figure 2: Excerpt of class diagram of MASEMUL.

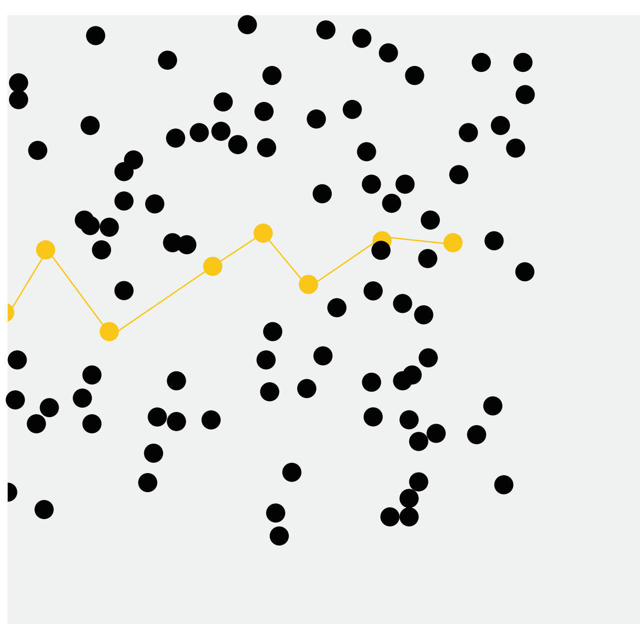


Figure 3: Example of simulation with sensor information of devices in MANETs (orange for highlighting the nodes of communication channels and black for the other nodes).

deployed, and one can observe the repercussion of one or several strategies. Each strategy will be represented with a different color selected by the user. In the simulation display, the channel including the devices and its links are highlighted with the color associated with the strategy. Among others, in each iteration, the simulator outputs (a) the number of broken channels as indicators of how many times the communication may be disrupted and (b) the average delays.

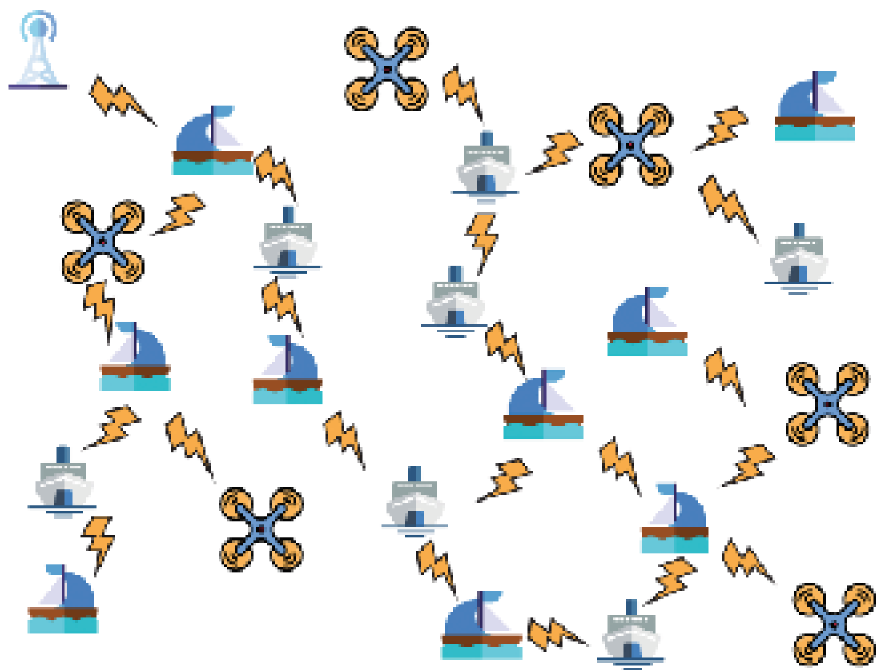


Figure 4: Network diagram of a maritime scenario.

Table 2: Parameters of the experimentation.

Parameter	Value
Simulated time	1000 s
Number of devices	50
Environment size (width of a square area)	1000 m
Maximum distance to communicate among two different devices	200 m
Maximum movement speed in each axis	5 m/s
Simulation speed (real time divided by simulated time)	10

The proposed approach is especially relevant in scenarios where there is enough network infrastructure and the communications rely on MANET nodes. Figure 4 shows a network diagram in a maritime scenario with different vehicles such as ships, sailboats, and UAVs that are moving. All these vehicles will know their location, direction, and speeds through GPS and digital compasses (also known as magnetometers). Vehicles communicate to other nearby vehicles conforming to a network using the defined scheduling strategy.

4. Experimentation

In order to show the potential of our novel simulator component, the proposed new simulator was compared with the well-known and validated 5G-air-simulator [16], in order to check that basic scenarios without movement information have obtained similar results. Table 2 shows the parameters that were used in all the experimentation as basis.

Although there are several testbeds for evaluating communications, to the best of the authors' knowledge, there is not any testbed that included the complete movement information (e.g., movement direction and acceleration) to validate the proposed simulation tool and movement-aware scheduling strategies.

Figure 5 compares the packet losses over the time simulated with the 5G-air-simulator and the packet losses obtained with MASEMUL. As one can observe, the packet loss results were similar in both simulators. More concretely, the average difference between the packet loss outputs of both simulators was only 2.77% in this experimentation. The packet losses are calculated in each time frame as the ratio of packets that have not arrived yet at their destination. It is worth noting that the packet delivery ratio is the opposite of the packet loss ratio used in these graphs, and consequently, the former can be easily calculated as 100% minus the information presented in these graphs. In the first time frames, the initial packets were sent and not received yet causing the effect of an apparently high packet loss percentage. This mainly shows a potential downside of this measurement mechanism rather than an actual network performance problem. Later, we have compared different scheduling algorithms considering location and movement information in some of them, discussing results. This shows the potential of movement-aware scheduling algorithms to improve QoS in the transmission of multimedia content in MANETs.

In order to test a new strategy referred to as "movement-aware" strategy in mobile networks, we compared this with a common heuristic strategy for obtaining pseudooptimized paths considering the distance as the prime factor denoted as "closest-to-target" strategy from this point forward. More concretely, each node will select the node that is closest to the target from the available ones nearby. This strategy is common in short-path routing protocols [19]. Then, the

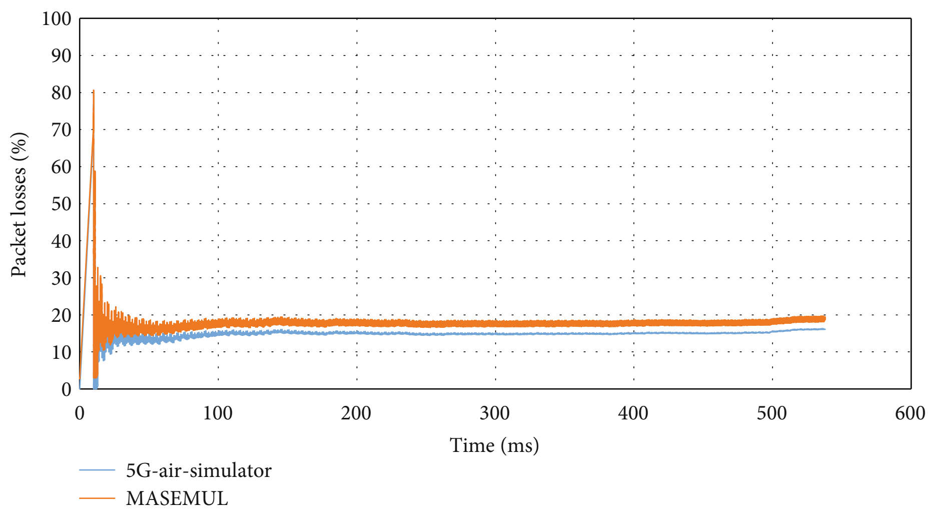


Figure 5: Packet losses simulated with 5G-air-simulator.

streaming channel will stay in the same channel of devices until this is broken, meaning that the distance between any two consecutive nodes surpasses the threshold of maximum possible distance for communication.

In the proposed movement-aware strategy, each node checks the relative movement over all the nodes in their available range. It will pick up the node that has the shortest heuristic considering the weighted sum of the square of the relative speed and the squared distance to the target. Equation (1) calculates the heuristic distance function that is used for selecting the next MANET node:

$$h\left(\vec{c}, \vec{m}_c, \vec{n}, \vec{m}_n, \vec{t}\right) = K_m * \left|\vec{m}_c - \vec{m}_n\right|^2 + K_d * \left|\vec{n} - \vec{t}\right|^2, \tag{1}$$

where h is the heuristic function, $\vec{c}$ is the current node location, $\vec{m}_c$ is the movement of the current node, $\vec{n}$ is the evaluated node location, $\vec{m}_n$ is the movement of the evaluated node, $\vec{t}$ is the target location, and K_m and K_d are the constants of the weighted sum for respective movement and the distance to the target.

These squared distances are efficiently calculated without the need of calculating any square root. The "Device" class has two methods for efficiently calculating the squared distance and squared relative speed with, respectively, Equations (2) and (3) without needing any square root:

$$\text{sqrDistance}\left(\vec{n}, \vec{t}\right) = \left(n_x - t_x\right)^2 + \left(n_y - t_y\right)^2, \tag{2}$$

$$\text{sqrRelativeSpeed}\left(\vec{m}_c, \vec{m}_n\right) = \left(m_{c,x} - m_{n,x}\right)^2 + \left(m_{c,y} - m_{n,y}\right)^2, \tag{3}$$

where all the vectors $\vec{n}$, $\vec{t}$, $\vec{m}_c$, and $\vec{m}_n$ have the same meaning as in the previous equations and where for any $\vec{v}$ vector, its components are represented as v_x and v_y.

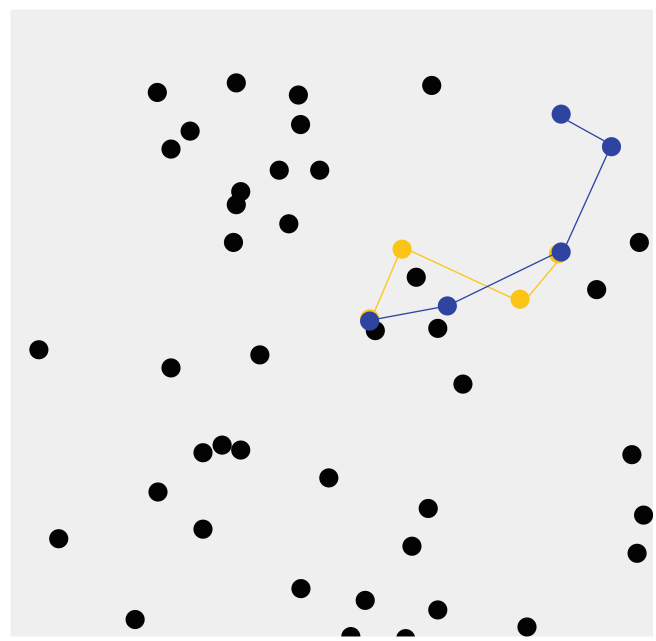


Figure 6: Closest-to-target (orange nodes and links) and movement-aware (blue nodes and links) strategies in the MANET (black for the other nodes).

Figure 6 shows an example of a simulated network scenario by using the novel MASEMUL simulator, which is comparing closest-to-target and movement-aware strategies, with, respectively, orange and blue colors. Notice that in the simulator, the devices continuously move and stay in the channel while this is not broken. In these experiments, we simulated an environment of 50 devices with the capacity of communicating over 200 m in a total area of 1 km × 1 km. The maximum speed was 5 m/s in each axis component, in which each simulation iteration represented a second of real-world time. In MASEMUL, it took 5386 ms of execution time to complete the simulation of the strategies.

Figure 7 compares the percentages of broken channels for the two strategies, considering the percentage of iterations in

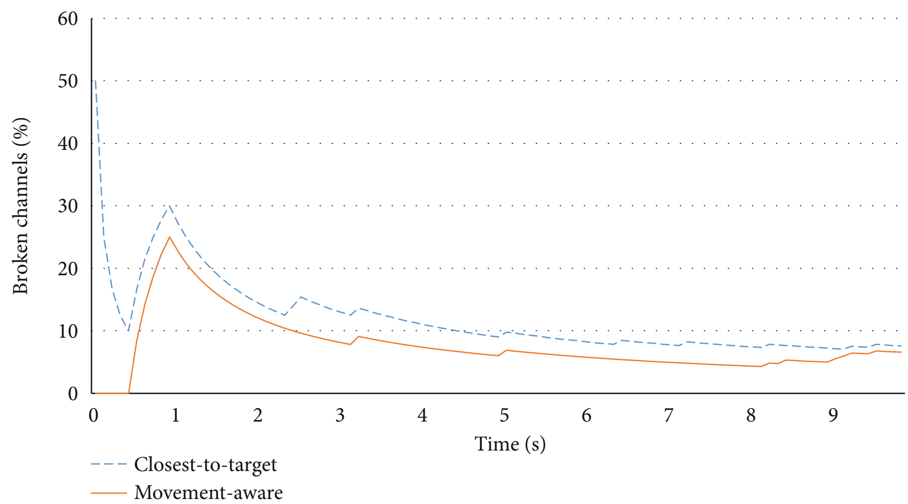
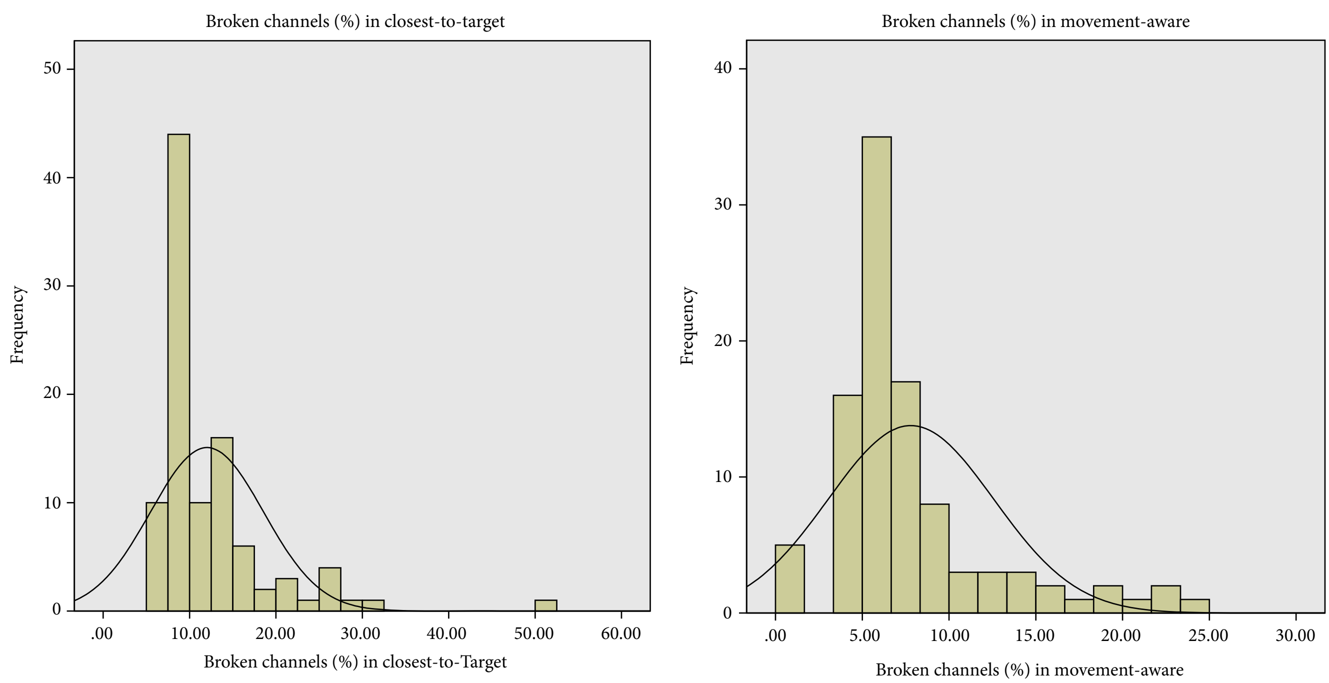


Figure 7: Comparison of broken channels.



(a) Closest-to-target strategy

(b) Movement-aware strategy

Figure 8: Histograms of broken channels.

which the channel needed to be reestablished. In addition, Figure 8 shows the histograms of broken channels for both strategies. One can observe that the movement-aware strategy had a lower number of broken channels in comparison with the closest-to-target strategy. In addition, the distribution of the percentages of broken channels is lower in the movement-aware strategy (i.e., with a range from 0.0% to 25.0%) than in the closest-to-target strategy (i.e., with a range from 5.0% to 52.5%). Multimedia traffic benefits from this lower percentage of broken channels to facilitate the seamless synchronous audio-visual communication for virtual meetings among the passengers of several vehicles of the MANETs.

Figure 9 compares the average delays between both strategies. In addition, Figure 10 shows the histograms of delays for both strategies. As one can observe, there is no clear advantage of one over the other, as in some iteration one is better and then the other way around. In general, the closest-to-target strategy gets the lower delays if there is no broken channel. However, broken channels also introduce some delays, and consequently, in these cases, then the movement-aware strategy had better results in some iterations. On average, closest-to-target had slightly lower average delays with 159 ms than with movement-aware strategy with a 171 ms of delay in average. In real-time traffic, delay is a crucial QoS parameter, although in this case this difference

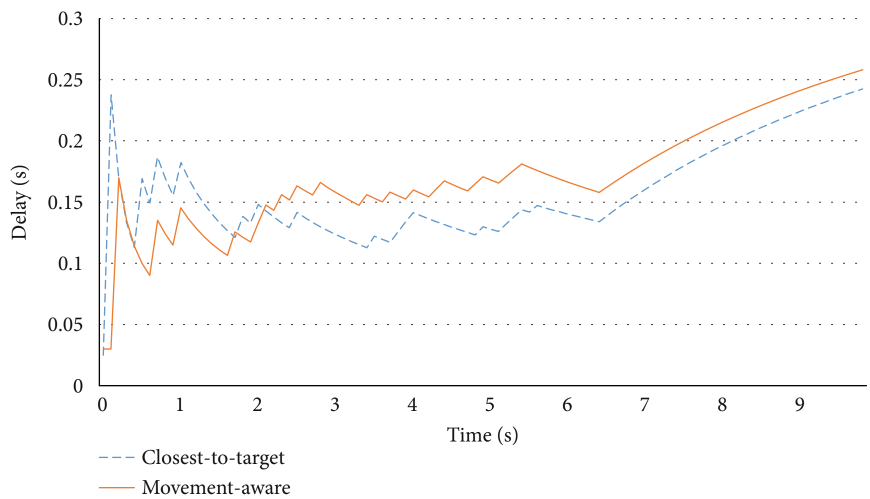
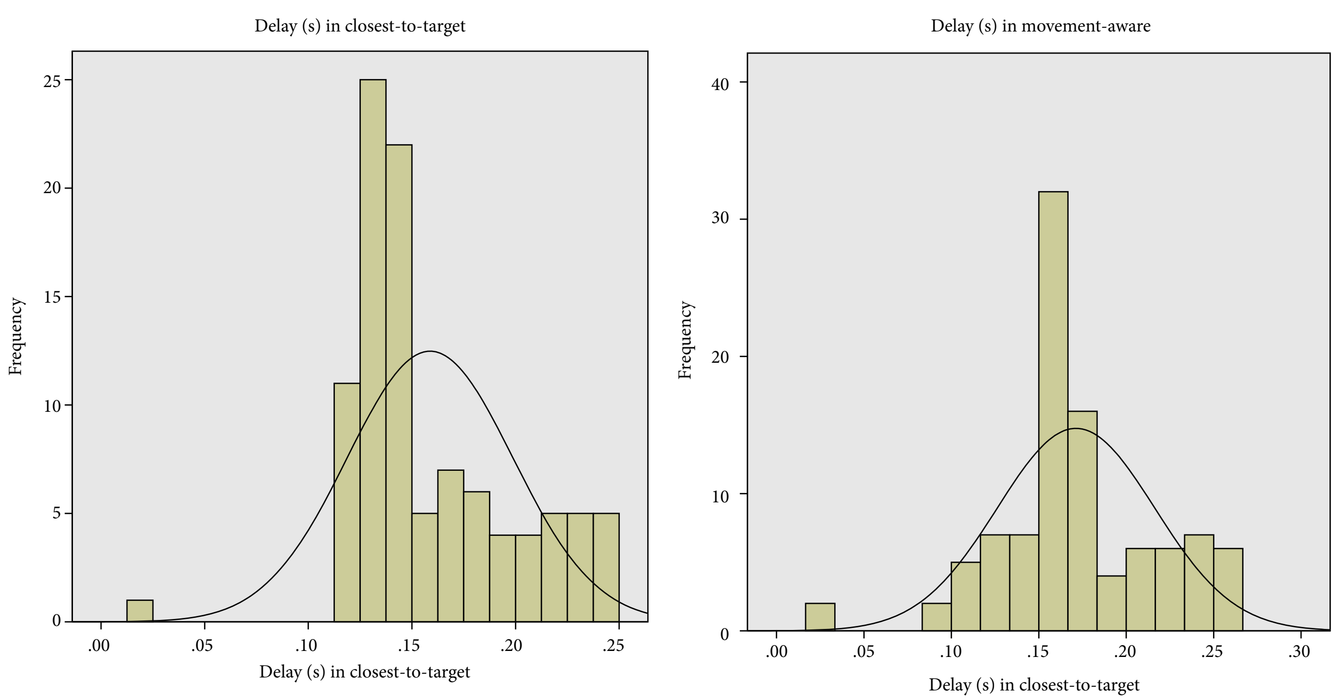


Figure 9: Comparison of delays.



(a) Closest-to-target strategy

(b) Movement-aware strategy

Figure 10: Histograms of delays.

can be considered negligible as it is about one hundredth of a second.

A strong point of the movement-aware strategy is that even though the delay is slightly higher with a low number of broken channels, the advantage is that when having a low number of broken channels, it will lead to having less packet losses and which will guarantee a better visual quality and seamless delivery of multimedia contents.

Furthermore, we have compared the performance of the proposed MASEMUL simulator with the most similar existing MANET simulator considering fair comparisons using the same platform and programming language. In particular, we considered that the most similar simulator is Yet Another MANET Simulator, as it is also specifically focused on MANETs with different movements of their nodes. It also uses the same programming language, i.e., Java. We executed both simulators with the same Java virtual machine, in the sample platform (i.e., Windows 10) and the same hardware (Lenovo Laptop p52 with Intel Core i7 vPro 8th generation processor and 16 GB of RAM). We used the Performance Monitor (also known as PerfMon) of Windows for measuring this comparison. Figure 11 shows the comparison of processor usage between Yet Another MANET Simulator and MASEMUL. It shows that MASEMUL improves the processor usage

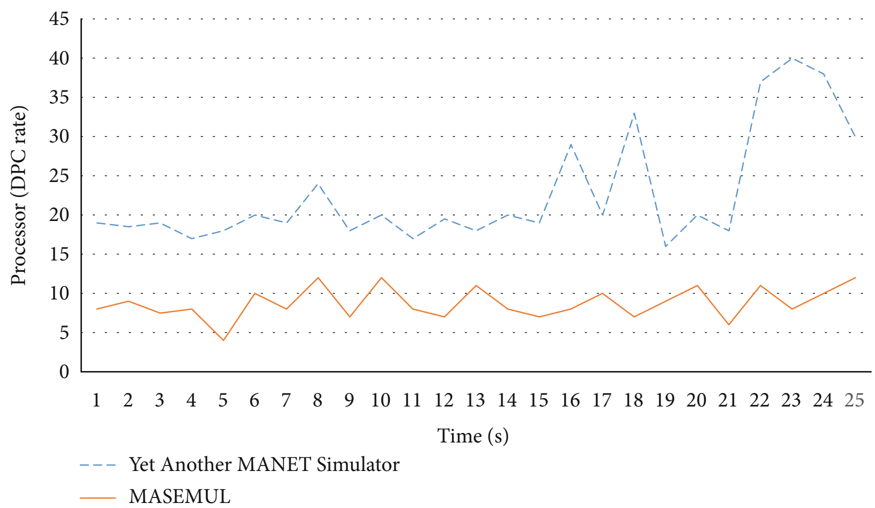


Figure 11: Comparison of processor usage.

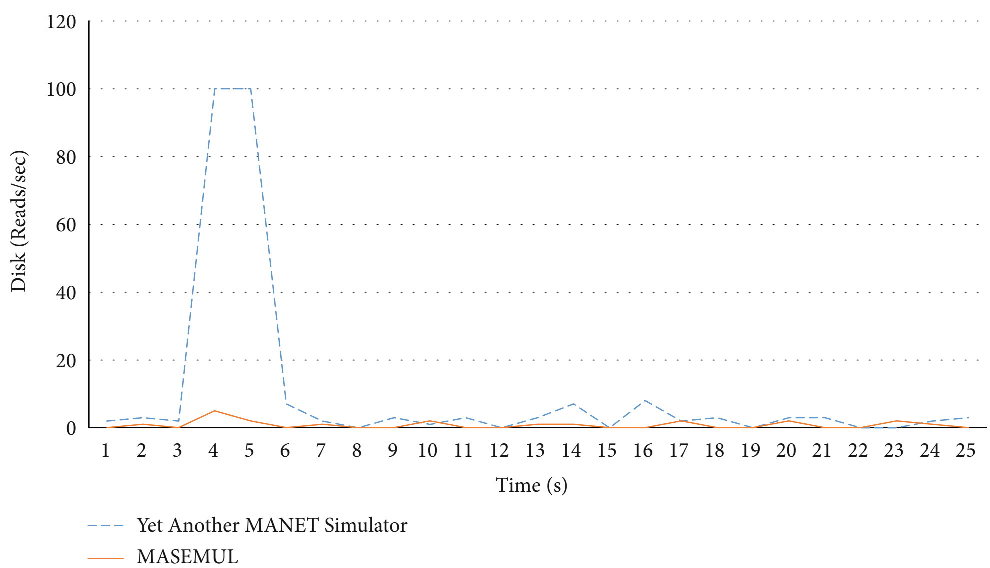


Figure 12: Comparison of disk usage.

decreasing it over the existing alternative Yet Another MANET Simulator. Figure 12 compares the disk usage. MASEMUL has a lower usage than Yet Another MANET Simulator, specially when launching the application at the beginning of the execution.

5. Discussion

In conclusion, the proposed MASEMUL has assisted in defining movement-aware scheduling strategies. The presented movement-aware strategy reduced the percentage of broken channels, which is useful for carrying real-time multimedia transmissions.

It is worth noting that in the movement-aware strategy when having a great weight in the part of relative movement speeds, it could obtain longer paths with a higher number of nodes, increasing the probability of a broken channel due to the higher number of devices that could separate from each other. Notice that both algorithms use greedy algorithms in which a decision of a path may imply that later in other points of the path conditions could be worse. Thus, only brute-force algorithms could guarantee the minimum relative speed considering all the nodes of the path, and their computational costs are not affordable even in medium MANETs. In brief, in the proposed movement-aware strategy, the calibration of heuristic weights is important to obtain the actual reduction of the percentage of broken channels.

Machine learning approaches have proven to be useful for estimating the quality of experience obtained by users in networks [15]. In addition, realistic simulators such as 5G-air-simulator provide ways of realistically simulating different network aspects [16]. The existing MANET simulator DEVS-M focuses on load balancing. However, none of these tools allow experimenting with the basics of simple MANET scheduling strategies that are movement-aware in networks of mobile devices for reducing the number of interruptions as the current work does, which helps to improve the QoS of multimedia transmission in MANETs.

Since the network is relying on information from device sensors, there is a risk of suffering perception-layer attacks in which the attacker can alter the information perceived by devices. In these attacks, the network would malfunction as the scheduling would be based on unreliable information. The proposed approach could be secured by identifying perception-layer attacks based on the detection of unusual patterns of location changes and movements as addressed in other perception-layer security mechanisms [20]. In particular, each device could use any of the available lightweight machine learning or novelty-detector approaches for this purpose. The MANET scheduling mechanism could also apply some general analysis to detect unusual patterns in the evolution of movements and locations of the devices to not only rely on the specific security measures of each device.

The goal of MASEMUL is to support scheduling strategies based on the movement information, and these movement-aware strategies cannot be directly implemented in other network simulators such as ns-2, ns-3, OMNet++, and OPNet. The goal of this work is not to develop a network simulator that replaces such existing network simulators but rather to actually complement these simulators with some new support for this new kind of movement-aware scheduling strategies. In fact, we plan to integrate MASEMUL functionality as a plugin of any of these network simulators.

6. Conclusions and Future Work

This work has presented a framework for simulating different scheduling strategies in MANETs considering that the moving devices have sensors that can track their movement direction and speed. In this way, new scheduling strategies can be defined considering this information. The novel simulator has used the 5G-air-simulator as a basis for checking its realistic simulations in some scenarios. The article has shown that a proposed movement-aware strategy has improved another strategy just aware of locations in mobile networks, in terms of percentage of broken channels over time. The former strategy is also aimed at improving the QoS of multimedia transmission.

As future work, we plan to develop new movement-aware strategies that are aware of context information for obtaining better quality parameters such as packet losses, delay, and throughput. We also plan to apply these strategies in real MANET scenarios for further assessing them.

Data Availability

This article mentions all the relevant aspects of the data involved in this research, and there is no other external resource with available data about this work.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors would like to acknowledge Prince Sultan University (PSU) and Smart Systems Engineering Lab for their valuable support and provision of research facilities that were essential for the completion of this work. This work is supported by the research project "Utilisation of IoT and sensors in smart cities for improving quality of life of impaired people" with reference 52-2020. We also acknowledge "CITIES: Ciudades inteligentes totalmente integrales, eficientes y sotenibles" (ref. 518RT0558) funded by CYTED Ciencia y Tecnología para el Desarrollo and "Diseño colaborativo para la promoción del bienestar en ciudades inteligentes inclusivas" (TIN2017-88327-R) funded by the Spanish council of Science, Innovation and Universities from the Spanish Government. This work has been partially supported by the "Ministerio de Economía y Competitividad" in the "Programa Estatal de Fomento de la Investigación Científica y Técnica de Excelencia, Subprograma Estatal de Generación de Conocimiento" within the project under Grant TIN2017-84802-C2-1-P. This work has also been partially supported by European Union through the ERANETMED (Euromediterranean Cooperation through ERANET joint activities and beyond) project ERANETMED3-227 SMARTWATIR.

References

[1] S. M. Adam and R. Hassan, "Delay aware reactive routing protocols for qos in manets: a review," *Journal of applied research and technology*, vol. 11, no. 6, pp. 844–850, 2013.

[2] E. Fki, I. Lahyani, I. Abdennadher, and R. Abid, "QoS-driven architectural mining for publish/subscribe systems deployed on MANET," *Procedia computer science*, vol. 112, pp. 260–269, 2017.

[3] Z. Han, A. L. Swindlehurst, and K. Liu, "Optimization of MANET connectivity via smart deployment/movement of unmanned air vehicles," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 7, pp. 3533–3546, 2009.

[4] Z. Chen, W. Zhou, S. Wu, and L. Cheng, "An adaptive on-demand multipath routing protocol with QoS support for high-speed MANET," *IEEE Access*, vol. 8, pp. 44760–44773, 2020.

[5] M. M. Nasralla, N. Khan, and M. G. Martini, "Content-aware downlink scheduling for LTE wireless systems: a survey and performance comparison of key approaches," *Computer Communications*, vol. 130, pp. 78–100, 2018.

[6] K. Venkatesh, N. Nithiyanandam, and Sivaneshkumara, "ANFIS based QoS-aware routing protocol for video streaming in MANETs," in *2019 IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS)*, pp. 1–6, Tamilnadu, India, April 2019.

[7] M. M. Nasralla, "A hybrid downlink scheduling approach for multi-traffic classes in LTE wireless systems," *IEEE Access*, vol. 8, pp. 82173–82186, 2020.

[8] G. Nardini, G. Stea, A. Virdis et al., "Practical feasibility, scalability and effectiveness of coordinated scheduling algorithms in cellular networks towards 5g," *Journal of Network and Computer Applications*, vol. 106, pp. 1–16, 2018.

[9] L. M. Tuan, L. H. Son, H. V. Long et al., "ITFDS: channel-aware integrated time and frequency-based downlink LTE scheduling in MANET," *Sensors*, vol. 20, no. 12, p. 3394, 2020.

[10] S. Song, Y. Liu, L. Guo, and Q. Song, “Optimized relaying and scheduling in cooperative free space optical fronthaul/backhaul of 5g,” *Optical Switching and Networking*, vol. 30, pp. 62–70, 2018.

[11] Z. Ma, B. Li, Z. Yan, and M. Yang, “Qos-oriented joint optimization of resource allocation and concurrent scheduling in 5g millimeter-wave network,” *Computer Networks*, vol. 166, p. 106979, 2020.

[12] V. Sharma, I. You, and R. Kumar, “Resource-based mobility management for video users in 5g using catalytic computing,” *Computer Communications*, vol. 118, pp. 120–139, 2018.

[13] D. C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne, “Blockchain for 5g and beyond networks: a state of the art survey,” *Journal of Network and Computer Applications*, vol. 166, article 102693, 2020.

[14] F. Çelik, “DEVS-M: a discrete event simulation framework for MANETs,” *Journal of Computational Science*, vol. 13, pp. 26–36, 2016.

[15] I. U. Rehman, M. M. Nasralla, and N. Y. Philip, “Multilayer perceptron neural network-based QoS-aware, content-aware and device-aware QoE prediction model: a proposed prediction model for medical ultrasound streaming over small cell networks,” *Electronics*, vol. 8, no. 2, p. 194, 2019.

[16] S. Martiradonna, A. Grassi, G. Piro, and G. Boggia, “Understanding the 5G-air-simulator: a tutorial on design criteria, technical components, and reference use cases,” *Computer Networks*, vol. 177, article 107314, 2020.

[17] G. Piro, L. A. Grieco, G. Boggia, F. Capozzi, and P. Camarda, “Simulating lte cellular systems: an open-source framework,” *IEEE Transactions on Vehicular Technology*, vol. 60, no. 2, pp. 498–513, 2010.

[18] I. García-Magariño, A. Gómez-Rodríguez, J. C. González-Moreno, and G. Palacios-Navarro, “PEABS: a process for developing efficient agent-based simulators,” *Engineering Applications of Artificial Intelligence*, vol. 46, pp. 104–112, 2015.

[19] O. Souihli, M. Frikha, and M. Ben Hamouda, “Load-balancing in MANET shortest-path routing protocols,” *Ad Hoc Networks*, vol. 7, no. 2, pp. 431–442, 2009.

[20] M. M. Nasralla, I. García-Magariño, and J. Lloret, “Defenses against perception-layer attacks on iot smart furniture for impaired people,” *IEEE Access*, vol. 8, pp. 119795–119805, 2020.