

# An Open Source and System-Level Simulator for the Internet of Drones

Giovanni Grieco\*, Roberto Artuso\*, Pietro Boccadoro\*<sup>‡</sup>, Giuseppe Piro\*<sup>‡</sup>, and Luigi Alfredo Grieco\*<sup>‡</sup>

\*Dept. of Electrical and Information Engineering (DEI), Politecnico di Bari, Bari, Italy,

<sup>‡</sup>CNIT, Consorzio Nazionale Interuniversitario per le Telecomunicazioni

Email: {giovanni.grieco, roberto.artuso, pietro.boccadoro, giuseppe.piro, alfredo.grieco}@poliba.it

**Abstract**—The Internet of Drones (IoD) represents a key enabling paradigm for the Internet of Mobile Things. By offering smart interactions among unmanned aerial vehicles, i.e., drones, it promises to support many advanced services, spanning from control signaling and traffic management to real-time applications and multimedia streaming. From the networking perspective, IoD should face heterogeneous architectures and extremely dynamics topologies, while ensuring the different levels of quality of service expected for the enabled applications. In this context, the definition of novel communication protocols and algorithms must be properly supported by valid instruments able to model the overall network architecture, as well as to evaluate and to improve its performance in many operating scenarios. To meet this goal, this work presents an open source and system-level simulator for the IoD. Developed on top of the well-known Network Simulator 3, it implements the key networking elements (drones, network access points, and Zone Service Provider), a standard-compliant communication stack based on the IEEE 802.11 technology, and scenarios with various mobility model. Thanks to its native modularity, the simulator is ready to be used for investigating the performance of customizable scenarios with a variable number of drones, as well as to be easily extended with new functionalities.

**Index Terms**—Internet of Mobile Things, Internet of Drones, system-level simulator, Network Simulator 3

## I. INTRODUCTION

While the Internet of Things (IoT) is continuously evolving, the scientific community is currently investigating new approaches to offer a seamless connectivity to mobile objects. Hence, new research frontiers are converging towards the so called Internet of Mobile Things (IoMT) [1]. In this context, the IoD is emerging as a key enabling paradigm for the IoMT. Conceived as a layered network control architecture for flying objects, with autonomous driving capabilities, namely Unmanned Aerial Vehicles (UAVs) or drones, the IoD promises to offer continuous, dynamic, distributed, and fine-grained control of the airspace [2]–[5], thus supporting a plethora of applications, spanning from public safety [6] to smart cities [7], and Intelligent Transportation Systems [7], [8]. In the context of disaster management, both military and civil drones may be involved [9]. While each of these application is evolving, the need for the massive employment of drones is promoting the advance functionalities [10], such as advanced mission scheduling,

and coordinated flight and collaboration between UAV forming swarms [11].

The development of the IoD networking paradigm is mandatory for drone-based civil and military applications and services [12], [13]. However, while the IoD tends to upgrade the current Internet environment in order to become more pervasive and ubiquitous, several challenging issues emerge from the communication point of view: IoD should face heterogeneous architectures and extremely dynamics network topologies, while ensuring the different levels of quality of service expected for the enabled applications [5], [14], [15]. Indeed, while novel communication protocols and algorithms should be properly conceived, the upcoming research activities will soon require the support of flexible tools able to investigate and optimize the performance of IoD solutions for various operating scenarios, before their actual deployment. Unfortunately, the current scientific literature does not offer any thorough answer to this need, yet.

To bridge this gap, this work presents an open source and system-level simulator for the IoD. Developed on top of the well-known Network Simulator 3 [16], it implements the main networking elements (drones, network access points, and Zone Service Provider (ZSP)), a standard-compliant communication stack based on the IEEE 802.11 technology (already available within the Network Simulator 3 framework), and four different scenarios, each one with its specific mobility model. Moreover, example scripts allows to simulate a variable number of drones that: move according to the mobility model associated to the reference scenario and exchange messages with network access points deployed on the Earth surface. The simulation generates log and PCAP files, useful to investigate the network behavior during the time and extract its performance. Thanks to its native modularity, the simulator is ready to be used for evaluating the communication performance in custom scenarios with a variable number of drones, as well as to be easily extended with new functionalities. The code is freely available at <https://telematics.poliba.it/iod-sim>.

The remainder of this work is as follows: Section II discusses possible application scenarios and introduces

related research issues. Section III describes the implemented simulator. Section IV presents the usage of tool through a concrete example. Section V concludes the work and draws future research activities.

## II. WHY DO WE NEED AN IOD SIMULATOR?

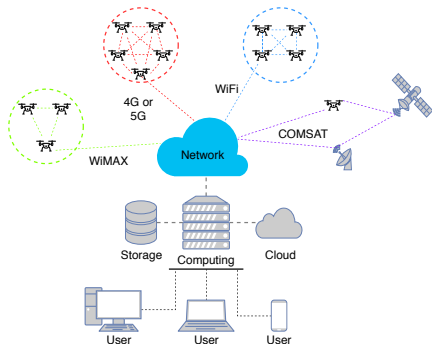


Fig. 1. High-level description of an IoD architecture.

The IoD can be defined as a heterogeneous and large-scale communication architecture that supports complex tasks management and coordination. Here, some of the challenges are related to airspace awareness, Non-Line of Sight (NLoS) and augmentative control and collision avoidance. The operational landscape and applicability of the IoD paradigm is described in Figure 1. Regardless of the reference application field, instead, a common concern regards wireless communication ranges: as long as they increase (i.e., over long distances) drones may become difficult to be controlled. More in general, the main questions that were formulated by the IoD researchers are summarized in what follows. What technological solutions are currently able to simulate communications between Drones? Are the protocol solutions we have today sufficient or do we need new ones? What are the limits of the applicability of the technological solutions currently available to this technological scenario? With which protocol solutions are we able to support the mobility that the drone expresses during its operation? Is this capacity actually supported even in realistic scenarios?

Turning from theory to practice, there is the need to validate both the concept and the modeling and design of the IoD. To this aim, a number of questions are still open for comments. For instance, it has still to be defined whether the system is autonomous or not. Moreover, the IoD has not been characterized in terms of network and service efficiency, yet. It is worth specifying that such an evaluation could be reviewed in strict correlation with the specific application and or operational condition. Another verification asset could be the overloading. For instance, which are the operational conditions and/or case in which the system does not withstand the data load. The investigation could be of relevance if it could

be possible to establish a correlation between the break-even point and the specific application/user. Further, a number of question are still left unspoken. In particular, in terms of telecommunications technologies, their integration should be evaluated as for their relevance and suitability to the typical IoD constraints.

The number and kind of open issues clearly suggest that there is the need for a dedicated platform that allows to evaluate different technological solutions. Hence, the preliminary assessments on the suitability of each of the proposals could be stated properly analyzing the results of the simulations conducted in functionally specified operational scenarios. All these considerations are of greater relevance in the IoD context, in which a simulator is highly desirable.

## III. THE DEVELOPED SIMULATION TOOL

The IoD simulator proposed herein has been developed on top of Network Simulator 3 [16], a well-known software solution widely conceived as a reliable open-source, discrete-event network simulator. The choice is motivated by the fact that, as a main design criteria, Network Simulator 3 proposes itself as a modular software solution modeling various communication protocols, while offering the possibility to be easily extended and enhanced. Moreover, it was licensed according to the GPLv2 licenses, which is a non-negligible pro as to be able to guarantee its diffusion within the research communities in the future.

Among the many proposed design challenges, it was first needed to characterize the drones' mobility model. For example, it had to be designed the number of directions a drone can follow. Nevertheless, moving is not only a matter of direction and it can be described relying on laws, models, and equations. The topic has been studied considering that the reference coordinates system is no longer a bi-dimensional one, but a 3D space. Consequently, concrete IoD scenarios where drones can move by following the aforementioned mobility models have been conceived. Then, the communication stack already available in the Network Simulator 3 has been harmoniously integrated within all the network elements. As a result, the proposed simulator already implements the fundamentals of data transmission and communication of a typical IoD scenario and can be easily extended in the future.

To make the simulation environment as portable as possible, the whole project has been released as a Docker container and the code is freely available at <https://telematics.poliba.it/iod-sim>.

To conclude, Figure 2 shows the Unified Modeling Language (UML) description of main classes involved in the simulator, whose functionalities will be described in what follows.

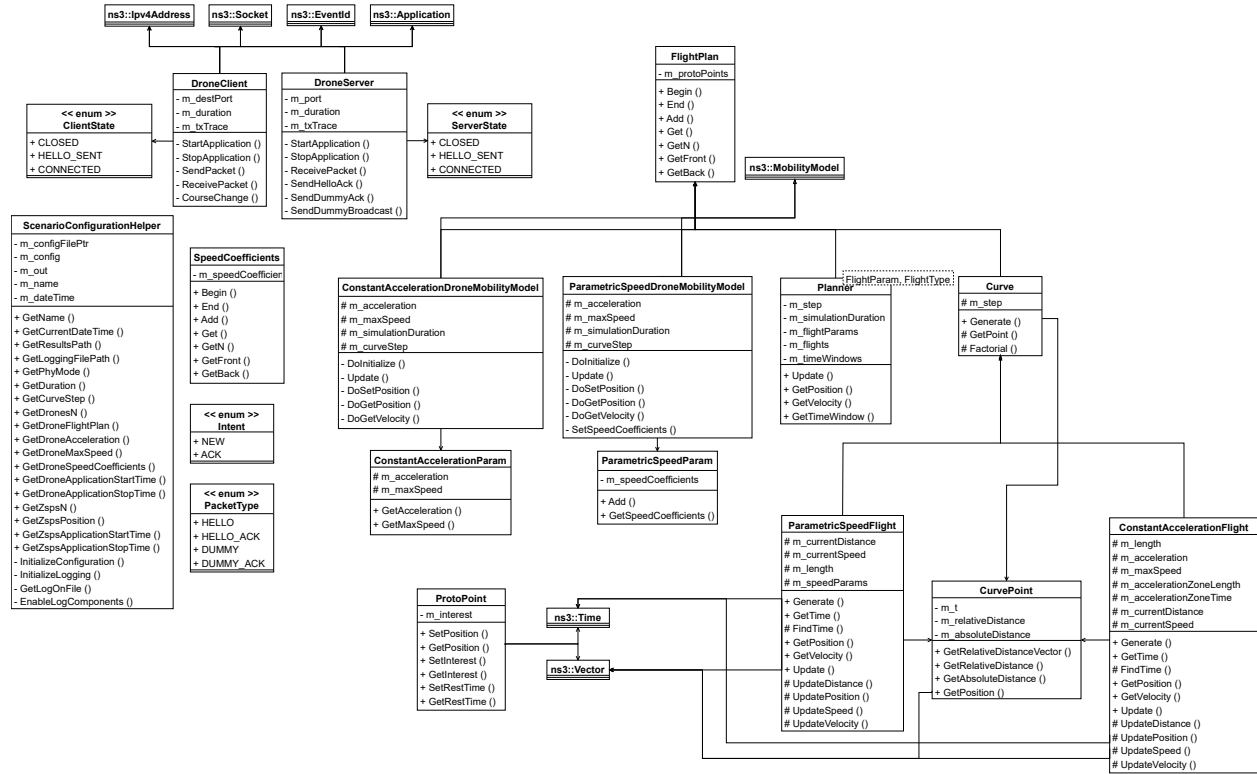


Fig. 2. UML diagram of the main classes involved in the implemented IoD simulator.

### A. Mobility Model

In this simulator, the flight of a drone has been modeled according to linear and non-linear trajectories. In the former case, the trajectory simply results from the interpolation of two consecutive waypoints. The assumption resulted in two mobility models:

- *Waypoint Mobility Model*: it allows to specify a number of waypoints that will be part of the flight plan of the drone. The trajectory will be linear from one point to the other.
- *Random Waypoint Mobility Model*: it extends the previous models by assuming that the waypoint list is generated with a uniform distribution. Once again, the trajectory is linear from one waypoint and the following one.

The latter case, instead, is referred to non-linear trajectories. In this case, trajectories have been realized as Bézier curves. Each point, stored in a *CurvePoint* object, is linked to the relative euclidean distance with the preceding point. Moreover, the object considers the absolute distance as in the sum of relative distances of each curve point starting from the origin of the curve. Lastly, it is used the parameter that generated such point using Bézier equation. Then, Bézier knots, also referred to as Proto Points, and the curve steps used to generate the trajectory are stored in a *FlightPlan* object. Moreover, multiple Flights can be sequentially

managed through a *Planner*, which can be configured by using auxiliary objects, Flight and Param. Each trajectory has its own time window that specifies its validity in time. The non-linear trajectories have been implemented through the following mobility models:

- *Constant Acceleration Mobility Model*: allows to simulate a uniformly accelerated drone motion. It is initialized by passing its auxiliary configuration object named *ConstantAccelerationParam*, embracing two mandatory properties: *acceleration* and *maxSpeed* (see Table I).
- *Parametric Speed Mobility Model*: the flight of drones is described in terms of speed through a couple of properties: *ParametricSpeedParam* and *SpeedCoefficients* (see Table II).

### B. Reference IoD scenarios

The main network elements modeled within the implemented IoD simulator are: drones, access point and ZSP. Drones and access point communicate each other through wireless links. For simplicity, the ZSP is directly connected to the access point.

Network elements can be deployed according to four possible scenarios. In *Scenario 1*, drones move according to the *Waypoint Mobility Model*, which foresees a set of preassigned waypoint positions used to define a trajectory. In *Scenario 2*, drones move according to the *Random Waypoint Mobility Model*. In particular, the

TABLE I  
ATTRIBUTES FOR CONSTANT ACCELERATION MODEL

Attribute	Type	Description
Acceleration	Double precision floating point number	Constant drone acceleration [m/s <sup>2</sup> ]
MaxSpeed	Double precision floating point number	Drone speed limit [m/s]
FlightPlan	ns3::FlightPlan	Set of points of interest
Simulation-Duration	Double precision floating point number	Simulation duration [s]
CurveStep	Single precision floating point number	Step of the curve being generated. Values near 0 mean more points to generate, hence an improved resolution

TABLE II  
ATTRIBUTES FOR PARAMETRIC SPEED MODEL

Attribute	Type	Description
Speed Coefficients	ns3::Speed-Coefficients	Coefficients of the polynomial velocity
FlightPlan	ns3::FlightPlan	Points of interest points
Simulation-Duration	Double precision floating point number	Simulation duration [s]
CurveStep	Single precision floating point number	Precision of generated curve

set of waypoints are generated according to a uniform distribution. Generated coordinates are included in a cubic 3-D space 1 km<sup>3</sup> wide. In *Scenario 3*: drones move according to the *Constant Acceleration Drone Mobility Model*. Finally, in *Scenario 4*, drones move according to the *Parametric Speed Drone Mobility Model*. It is important to note all the aforementioned IoD scenarios can be configured by the user. The list of input parameters is summarized in Table III.

### C. A bottom-up description of the protocol stack

The current version of the IoD simulator integrates a standard compliant communication stack, based on the IEEE 802.11 specifications. The network level is based on Internet Protocol version 4 (IPv4). Moreover, to guarantee the effective transport of information and data, the transport layer of the protocol stack employs the User Datagram Protocol (UDP) protocol. The choice is motivated by the fact, that in the IoD context, high communication speed could be privileged instead of ensuring that every single packet is received. And, it is widely known that the UDP protocol is used by applications requiring high speed communication techniques.

TABLE III  
PARAMETER SETTINGS OF IMPLEMENTED IOD SCENARIOS.

	Attribute	Type	Description
<b>Scen. 1</b>	numDrones	uint32	Number of drones
<b>Scen. 2</b>	numDrones	uint32	Number of drones
	duration	double	Simulation duration
<b>Scen. 3</b>	acceleration	double	Drone acceleration
	maxSpeed	double	Drone max speed
	applicationStartTime	double	Start time
	applicationStopTime	double	Stop time
	trajectory	Array of Objects	Drone trajectory
	logOnFile	bool	Enable file logging.
	phyMode	string	Transmission mode
	duration	double	Simulation duration
	drones	Array of Objects	Drones properties
	Zone Service Provider	Array of Objects	Zone Service Provider configuration
logComponents	Array of Strings	Log enable	
<b>Scen. 4</b>	applicationStartTime	double	Start time
	applicationStopTime	double	Stop time
	speedCoefficients	Array of double	Velocity coefficients
	trajectory	Array of Objects	Drone trajectory
	logOnFile	bool	Enable file logging
	phyMode	string	Transmission mode
	duration	double	Simulation duration
	drones	Array of Objects	Drones properties
	Zone Service Provider	Array of Objects	Zone Service Provider configuration
	logComponents	Array of Strings	Log enable

### D. A customized client-server application

At the application layer, the interaction among drones and access points has been modeled according to the typical client-server pattern. The two communicating peers are: *ns3::DroneClient* and *ns3::DroneServer*, installed on drones and network ZSP, respectively.

To assess network initial state at simulation startup, a rendezvous process is triggered in order to auto-discover network clients (i.e., drones) and the server (i.e., ZSP).

Communication is simulated in an event-driven fashion. In particular, the models implements dedicated Application Programming Interfaces (APIs) and manages packet transfer via specialized methods, such as *SendPacket()* and *ReceivePacket()*. The application message has been designed to encode data using the JavaScript Object Notation (JSON), which is then encapsulated and transported as an UDP payload. Note that JSON format files require a simple structure with a minimum number of tags.

The JSON message contains the following fields: drone identified (32 bit), a packet sequence number (32 bit), a command field (string), the drone localization in-

formation (object). Moreover, latitude (64 bit), longitude (64 bit), altitude (64 bit) and speed (32 bit) are included too. The command (cmd) field is used to describe what the message content and how it should be interpreted. Such values can be set to *HELLO*, *HELLO\_ACK*, *UPDATE* and *UPDATE\_ACK*.

The former couple is used for the rendezvous process. In particular, the *HELLO* message is sent by the drone in broadcast. The reply, that is the *HELLO\_ACK*, is sent by the ZSP to the drone. A typical example of this communication are related to the transmission of the Global Position System (GPS) signal, which offers drone geographic coordinates and its velocity in a structured way. Such coordinates are relative to the simulated space, which can be assumed a 3-D space.

*UPDATE* and *UPDATE\_ACK* packets, instead, are used in the periodic data exchange between drones and ZSPs. In details, a drone periodically sends a *UPDATE* and gets the related *UPDATE\_ACK* from the ZSP, which represents the confirmation of receipt of the message. As a design criteria, drones are, de facto, conceived as data producer data. This is because their monitoring activities implies that, once they reach the points of interest, they are suppose to carry out some task, such as monitor environmental parameters or taking pictures of a desired place. To this aim, the transmission period and the overall amount of packets to be sent have been implemented as drones properties and can be set in each and every scenario.

### E. Log files

During the simulation, three types of log files are generated: summary-scenario, PHY log, and PCAP file. First, the summary-scenario file contains information on the simulated scenario, together with date and time of execution. This log also presents useful information about the number of drones, the number of ZSPs and the total duration of the simulation. The PHY log records all the simulation details related to the message exchanged at the physical layer, during the time. Each line of the file contains: a marker  $t$  or  $r$ , to indicate if the frame is transmitted or received, a timestamp, the unique identifiers of the involved network elements and other information related to the simulation of the involved objects. The PCAP file records exchanged packets and their details at different levels of the protocol stack. Specifically, for each exchanged packet, the PCAP file contains a timestamp, an indication of the data source, an indication of the receiver of the message, the reference protocol for the message, the length of the message (expressed in bytes), and the message itself. This is useful for the analysis of network functionality and the diagnosis of any problems encountered during transmissions.

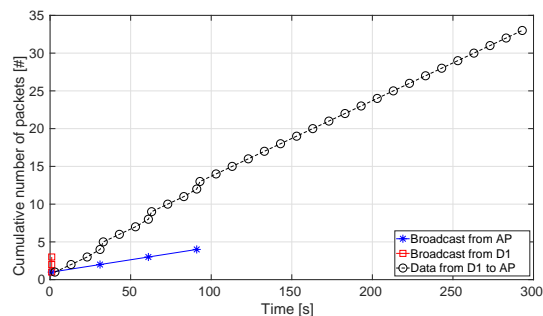


Fig. 3. Cumulative number of packets exchanged throughout the simulation.

## IV. USING AND EXTENDING THE IOD SIMULATOR

### A. Simulation example

The concrete usage of the implemented IoD simulator is herein presented. The simulation example is based on Scenario 2. The overall simulation duration has been set to 300 seconds. During this period of time, the scenario foresees 1 ZSP and 9 drones moving all around a certain area. During the simulation, drones move according to the random position model; in this case, in fact, drones are moving from one point to another in a straight way. The speed of a drone is not constant and can be calculated through its definition, hence as the ratio between the final and starting positions over the reference time, indicated through the respective timestamps. At the beginning of the simulation, the ZSP is communicating with drones by sending them broadcast messages at a fixed pace, i.e., every 30 seconds. These messages are related to rendezvous process, which happens in the startup phase. Drones periodically send data to the ZSP, e.g., every 10 seconds. The PHY log file has been properly processed in order to generate Figures 3 and 4. In details, Figure 3 shows the cumulative amount of packets sent by a one single drone over time. In this graph it is clearly reports an initial messages exchange between the drones and the ZSP. During the first seconds of the simulation, the drones generates broadcast traffic. After the startup phase is concluded, the simulation continues with a periodic message exchange that fits the aforescribed settings. Figure 4, instead, highlights the impact of packet loss (due to drones' mobility) on network performance. Further, it is worth noting that the conducted simulation refer to communications that are occurring without losses. Nevertheless, the mobility, together with channel access policy and physical losses lead to a reduction in performance.

### B. The road ahead

Future research efforts will be devoted to enhance simulator functionalities. In particular, the presented tool will be extended to support for swarm flight coordination, thus investigating collision avoidance solutions,

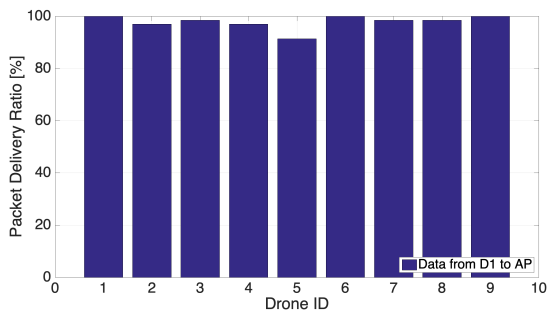


Fig. 4. Packet Delivery Ratio for each drone in the network.

with specific reference to Particle Swarm Optimization and Genetic Algorithm [17]. As for communications, the simulator will soon support a larger number of ZSPs. The communication protocols will include routing protocols for UAVs [18]. The presented module can be extended integrating graphical (i.e., *NetAnim*<sup>1</sup>) and data visualization tools such as (i.e. *gnuplot*<sup>2</sup> or *pyviz*<sup>3</sup>). The simulator will also support Python bindings, as to easily develop the upcoming scenarios. Lastly, the simulator could be used to evaluate security and privacy issues [19].

## V. CONCLUSION

The present contribution proposed an open source and system-level simulator for the IoD. The tool has been developed on top of the well-known Network Simulator 3 and foresees a number of networking elements, a standard-compliant communication stack, and four different scenarios, each one with its specific mobility model. The proposal has been proven to be modular by design, which implies its readiness for evaluating communication performance in custom scenarios with a variable number of drones. Further, it could be easily extended with new functionalities and protocol stack support. In the future, the simulator will be extended in order to evaluate and include swarm flight coordination, multiple ZSPs, and energy consumption models.

## ACKNOWLEDGMENT

This work was partially founded by Italian MIUR PON projects Pico&Pro (ARS01 01061) AGREED (ARS01 00254), FURTHER (ARS01 01283), RAFAEL (ARS01 00305) and by Apulia Region (Italy) Research Projects E-SHELF (OSW3NO1) and INTENTO (36A49H6).

<sup>1</sup><https://www.nsnam.org/wiki/NetAnim>

<sup>2</sup><https://www.nsnam.org/docs/manual/html/gnuplot.html>

<sup>3</sup><https://www.nsnam.org/wiki/PyViz>

## REFERENCES

- [1] K. Nahrstedt, H. Li, P. Nguyen, S. Chang, and L. Vu, "Internet of mobile things: Mobility-driven challenges, designs and implementations," in *IEEE First International Conference on Internet-of-Things Design and Implementation (IoTDI)*, Apr. 2016, pp. 25–36.
- [2] M. Gharibi, R. Boutaba, and S. L. Waslander, "Internet of drones," *IEEE Access*, vol. 4, 2016.
- [3] N. H. Motlagh, T. Taleb, and O. Arouk, "Low-altitude unmanned aerial vehicles-based internet of things services: Comprehensive survey and future perspectives," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 899–922, Dec. 2016.
- [4] T. Long, M. Ozger, O. Cetinkaya, and O. B. Akan, "Energy neutral internet of drones," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 22–28, Jan. 2018.
- [5] S. Dhingra, R. B. Mada, A. H. Gandomi, R. Patan, and M. Daneshmand, "Internet of things mobile - air pollution monitoring system (iot-mobair)," *IEEE Internet of Things Journal*, pp. 1–1, 2019.
- [6] X. Li, D. Guo, H. Yin, and G. Wei, "Drone-assisted public safety wireless broadband network," in *2015 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, Mar. 2015, pp. 323–328.
- [7] H. Menouar, I. Guvenc, K. Akkaya, A. S. Uluagac, A. Kadri, and A. Tuncer, "Uav-enabled intelligent transportation systems for the smart city: Applications and challenges," *IEEE Communications Magazine*, vol. 55, no. 3, pp. 22–28, Mar. 2017.
- [8] M. A. Khan, W. Ectors, T. Bellemans, D. Janssens, and G. Wets, "Uav-based traffic analysis: A universal guiding framework based on literature survey," *Transportation Research Procedia*, vol. 22, pp. 541 – 550, 2017, 19th EURO Working Group on Transportation Meeting, EWGT2016, 5-7 September 2016, Istanbul, Turkey.
- [9] A. Restas, "Drone applications for supporting disaster management," *World Journal of Engineering and Technology*, vol. 3, pp. 316–321, 2015.
- [10] A. Koubaa, B. Qureshi, M. Sriti, Y. Javed, and E. Tovar, "A service-oriented cloud-based management system for the internet-of-drones," in *2017 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, 2017, pp. 329–335.
- [11] A. Koubaa, B. Qureshi, M.-F. Sriti, A. Allouch, Y. Javed, M. Alajlan, O. Cheikhrouhou, M. Khalgui, and E. Tovar, "Dronemap planner: A service-oriented cloud-based management system for the internet-of-drones," *Ad Hoc Networks*, vol. 86, pp. 46 – 62, 2019.
- [12] A. Koubaa, B. Qureshi, M. F. Sriti, Y. Javed, and E. Tovar, "A service-oriented cloud-based management system for the internet-of-drones," in *IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, 2017, pp. 329–335.
- [13] M. A. Ma'sum, M. K. Arrofi, G. Jati, F. Arifin, M. N. Kurniawan, P. Mursanto, and W. Jatmiko, "Simulation of intelligent unmanned aerial vehicle (uav) for military surveillance," in *International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, Sep. 2013, pp. 161–166.
- [14] V. Sharma, G. Choudhary, Y. Ko, and I. You, "Behavior and vulnerability assessment of drones-enabled industrial internet of things (iiot)," *IEEE Access*, vol. 6, pp. 43 368–43 383, 2018.
- [15] J. Liu, H. Shen, H. S. Narman, W. Chung, and Z. Lin, "A survey of mobile crowdsensing techniques: A critical component for the internet of things," *ACM Trans. Cyber-Phys. Syst.*, vol. 2, no. 3, pp. 18:1–18:26, July 2018.
- [16] G. Carneiro, "Ns-3: Network simulator 3," in *UTM Lab Meeting April*, vol. 20, 2010, pp. 4–5.
- [17] X. Li, T. Zhang, and J. Li, "A particle swarm mobility model for flying ad hoc networks," in *2017 IEEE Global Communications Conference (GLOBECOM)*, Dec. 2017, pp. 1–6.
- [18] J. Jiang and G. Han, "Routing protocols for unmanned aerial vehicles," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 58–63, Jan. 2018.
- [19] C. Lin, D. He, N. Kumar, K. R. Choo, A. Vinel, and X. Huang, "Security and privacy for the internet of drones: Challenges and solutions," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 64–69, Jan 2018.