Smart City IoT On-Demand Monitoring System Using a Drone Fleet

Gordana Jotanovic[1](✉), Vladimir Brtka[2], Jelena Stojanov[2], Zeljko Stojanov[2], Goran Jausevac[1], and Dalibor Dobrilovic[2]

[1] Faculty of Transport and Traffic Engineering, University of East Sarajevo, Doboj, Bosnia and Herzegovina
{gordana.jotanovic,goran.jausevac}@sf.ues.rs.ba
[2] Technical Faculty "Mihajlo Pupin", University of Novi Sad, Zrenjanin, Serbia
vbrtka@tfzr.uns.ac.rs, {jelena.stojanov,zeljko.stojanov, dalibor.dobrilovic}@uns.ac.rs

Abstract. This paper deals with the management of the Drone Fleet in the areas of Smart Cities that are not infrastructurally covered by continuous monitoring systems. The proposed IoT architecture of the system is based on the cloud and implies the existence of Vehicle Detection Sensors. The role of the drone navigation service is especially elaborated. The Smart City area is divided into sectors, and the value of the potential load of the sector (PSL) is proposed as a measure of the traffic load of the sector. The use of the n-neighborhood concept enables the selection of critical sectors and the implementation of an algorithm that controls the movement of drones. The performed simulation of drone movement, by varying the number of drones and the number of sectors, indicates a change in the length of the distance traveled and the time required to visit all critical sectors. Procedures based on the n-neighborhood concept tend to be general and insensitive to the dynamic nature of traffic, as the set of critical sectors changes.

Keywords: Smart city · IoT · On-demand monitoring · Drone fleets · Vehicle detection sensors · N-neighborhood · Cloud-based drone service

1 Introduction

The use of Unmanned Aerial Vehicles (UAV, a.k.a. drones) has increased in recent years in terms of the number of drones and the number of roles and tasks that drones perform. The application domains cover military, delivery, monitoring, control and recording, to name a few. Also, new fields of application are opening. At the same time, traffic-related problems are becoming significantly more complicated, especially motor vehicle traffic within urban areas. Traffic problems are solved by available methods, but the results of the solutions are not always satisfactory or good enough.

This paper deals with the use of drones in order to improve the quality of monitoring in the areas of Smart Cities, as well as the possibilities of managing and using the

D. Perakovic and L. Knapcikova (Eds.): FABULOUS 2022, LNICST 445, pp. 105–121, 2022.
https://doi.org/10.1007/978-3-031-15101-9_8

Drone Fleet for on-demand monitoring. The use of a Drone Fleet usually involves three or more drones engaged in the task, while fleet management is done from the control center by applying artificial intelligence methods, i.e., machine learning. This research relies on the results of previous research that has dealt with the movement of drones along predetermined routes, which cover part of the urban area, but here more general case of Drone Fleet management is considered, so that there are no predetermined routes [1, 2]. The approach shown here implies that the urban area is divided into sectors, as well as the existence of the possibility of vehicle detection. Also, there is a need for the existence of a Control Center. The problems of calculating total sector traffic load and directing drones to selected sectors (activation of the sector) are discussed. The functions of drones within the sector were discussed in the previous works, and they can be simply positioning the drone camera and recording the situation or touring the sector along a predetermined route. This would avoid several problems related to the need for constant "patrolling", although the need for the existence of vehicle detection sensors increases the cost of the proposed solution.

The rest of the paper is structured as follows: The second part describes the results of selected papers that are relevant to the presented research in terms of topic, subject and problem. Selected previous works mainly deal with the use of drones in urban areas, drone management problems and route calculations. The third part describes the proposed design of a cloud-based IoT layered architecture for on-demand monitoring. The fourth part deals with the analysis of the urban area from the point of view of the Drone Fleet operation in the urban environment. This section presents the theoretical basis for simulating the movement of a Drone Fleet. The fifth chapter presents the results of research (simulations) and discusses them. The results are analyzed, and some potential problems are considered. The concluding part summarizes the results and gives possible directions for future research and upgrades of the proposed solution.

2 Previous Work

Cloud based systems have been intensively used for integrating and managing various ubiquitous systems because they enable services availability for variety of users from anywhere. The use of unmanned aerial vehicles (UAVs), commonly named drones, has expansive growth recently, while estimations indicate that UAV market will surpass USD 28.27 Billion by 2022 [3]. Majority of commercial UAVS use point-to-point communication between the drone and ground stations by using WiFi and TCP protocols, which put limitations on communication range, especially in large scale environments such as cities or larger areas. Use of cloud-based systems for managing drones is recent solution that may help in overcoming communication problems and increase the efficiency and availability of these systems. Cloud based systems with variety of services for analytics, computation, storage, and visualization are well suited for this purpose [4].

Koubâa et al. [3] presented Dronemap Planner (DP), a service-oriented cloud-based system for managing drones. DP enables control, monitoring and communication with drones over the Internet. This solution aims in overcoming limitation of drones related to computation and energy resources, that prevent them to run heavy application onboard. Cloud system provides computing resources to drones, virtualizes access to drones through Web services, schedules drones' missions and support communications among them. DP software architecture contains the following layers or subsystems that contain loosely coupled software components: (1) Communication layer (network sockets and web sockets components), (2) Proxy layer (components for protocol-related operations for message parsing, dispatching, and processing), (3) Cloud layer (with software components Cloud Manager, Storage, Web Services, and Cognitive intelligence), (4) Drone layer (information related to drones), and (5) User layer (components that enable users to access cloud system through web services). Cognitive Engine (CE) is a software component based on artificial intelligence techniques for planning drone paths and solving problems. DP evaluation was performed using real drone for real-time tracking application DroneTrack for tracking moving objects through cloud [5].

Capello et al. [6] proposed a cloud-based supervision system for Remotely Piloted Aircraft Systems (RPAS) in urban environments. The system architecture contains five connected layers: (1) Map Generation Layer – reliable for creating a dynamical map that assists in assessing the risk in the navigation area, (2) Path Planning Layer – reliable for computing a path for RPAS by considering a risk map produced by Map Generation Layer, (3) Control System Layer – reliable for sending commands to hardware layer, performing diagnosis and detecting failures (4) On-board Control System Layer – reliable for flight safety in any condition (it is activated only if a bad connection to cloud is observed), and (5) Hardware Layer – reliable for receiving the control commands and actuating the control devices based on the received signals from the Control layers. Innovative elements in the proposed system are the definition of a dynamic risk-map, and on-board control system for performing emergency maneuvers when the communication with the cloud is critical.

Hu et al. [7] presented CloudStation, a cloud-based ground control station software that enables communication between pilots and drones through TCP and HTTP protocols. CloudStation software provides a graphical user interface for pilots that can control drones. CloudStation architecture contains three parts: (1) Web browser for accessing web application for controlling drones, (2) Linux server in the cloud, and (3) Linux companion computer and a flight controller running Ardupilot on the drone. The software at Linux cloud server is implemented by using with Django, a Model-View-Controller web framework.

Mehrooz et al. [8] proposed a cloud-based platform for open-source Internet of Drones (IoD) application. The platform is based on distributed cloud service data infrastructure, Service-Oriented Architecture (SOA), and Robot Operating System (ROS). The proposed solution has three layers: cloud services, drone computing unit, and drone flight simulator or real time drone flight controller depending on the use. Cloud services layer provides computing software application, storage and network infrastructure. Overall system architecture is modelled in UML and contains four packages related to cloud services, ROS internal communication, flight simulator and real flight control. Cloud services uses REST API to send navigation information relevant for flight (GPS coordinates, position of power towers, etc.) to drones. Test scenario for the proposed cloud-based infrastructure is generated and executed in simulation by using the AirSim/Gazebo simulator.

In [9], dynamic graph convolutional network was considered to predict traffic-flow. Reinforcement learning was used, and the illustration of the application is based on the use of real data on the movement of bicycles. Although the paper does not deal with the use of drones, it is very useful from the point of view of modeling dynamic traffic flows and the methods that were used. The paper contains a description of how to generate dynamic graphs in order to model traffic flows. Of particular interest is "the research of using Euclidean space to model traffic networks", where "the stations are often alone or aggregated in a two-dimensional grid structure, which obviously ignores the topology information between the stations". The above two statements give an idea of the possible way to generate a model of an urban environment or part of an urban environment, and the structure of the model has an impact on the definition of algorithms for drones belonging to the same fleet. The idea of transforming Euclidean space into a non-Euclidean (graph) form is especially considered.

In [10] a time-varying and price-sensitive drone fleet queueing model was formulated for delivery tasks. An approximation algorithm was designed to numerically tackle the problem of Quality of Service, and simulations were performed. The stochastic optimization model was constructed in order to maximize the overall profit, constrained by system performances. This research gives some insight to optimization problems regarding drone fleet performance, as well as the number of drones within the fleet.

Research presented in [11] is more specific and deals with the use of drones for medical item delivery from a logistical point of view. The timing of delivery is critical, and the problem is solved by selecting locations for charging stations, assigning clinics to providers, and scheduling and sequencing the trips. Experiments were conducted, and the number of drones was varied to gain insight into the performance of the solutions proposed. A timeslot trip scheduling formulation is proposed and evaluated. This research points out importance of choosing drone battery recharging station locations or selecting flight starting points for each of the drones. Also, a hint of the idea of an algorithm for drone movement can be gained.

In [12] is given a description of a research on a drone fleet management for providing connectivity to sensors and actuators in Industrial Internet of Things (IIoT) scenarios. The Reinforcement learning (RL) algorithm was applied in order to achieve optimal drone management, deciding the number of drones and taking care of bandwidth. However, it is assumed that it is possible to constantly charge drone batteries, which greatly simplifies the control problem.

Research presented in [13] deals with traffic optimization. It is stated that current Artificial Intelligence (AI) technologies have difficulty in adapting to the dynamic nature of traffic network. Graph Neural Network (GNN) is proposed in order to model and optimize traffic in datacenters. GNN can provide accurate estimation of never-seen network states, while the generalization enabled by GNN overcomes the difficulty of adapting to the dynamic nature of traffic network.

An overview of literature references enables gaining insight into the subject and highlights the characteristics of existing solutions and problems. Based on that, the architecture of the system and the way of managing the drone fleet is proposed in the following sections.

3 IoT Architecture for On-Demand Monitoring

In smart cities, the urban areas that need to be monitored are large. Therefore, it is necessary to provide infrastructure support or use many monitoring devices. The price and number of devices for continuous monitoring, as well as resources for data storage and processing are a significant expense in monitoring systems. The paper proposes an IoT system for monitoring only areas of interest. The system is designed to save data storage capacity and processing resources and optimize the number of drones required for monitoring. The Smart City IoT on-demand monitoring system operates so that Drone Fleet cover the areas that are not covered by infrastructure for continuous monitoring systems. The system is Cloud-Based IoT architecture. The proposed IoT architecture is defined through five layers: three horizontal layers and two vertical layers [4]. The horizontal layers are: Perception Layer, Network Layer, Service & Application Layer (Fig. 1).

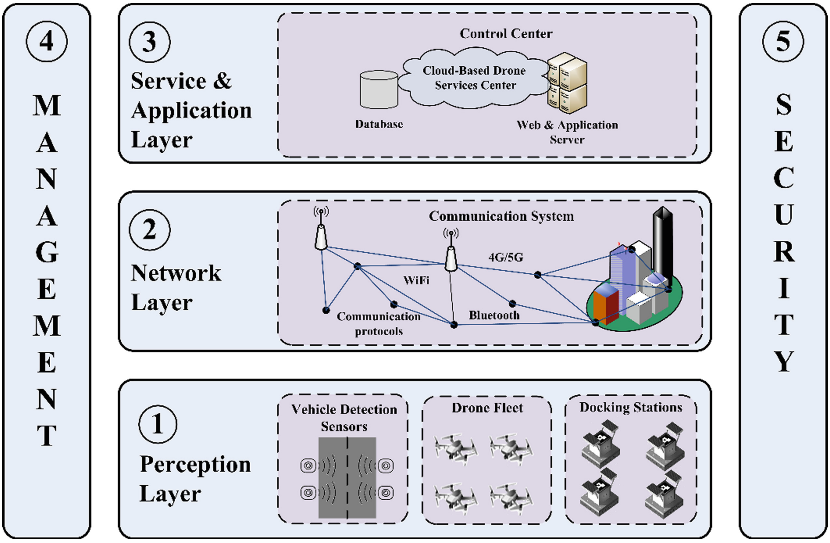


Fig. 1. IoT architecture for on-demand monitoring

The two vertical layers are Management and Security, which are responsible for the control and security of the entire proposed IoT system [14].

3.1 Perception Layer

The basic function of the Perception Layer is to collect data to identify specific parameters for each sector, such as location images, number of vehicles on roads in a defined time interval (traffic frequency), traffic noise, temperature, humidity and air pollution [15]. The perception layer of the IoT system presented in this paper contains Vehicle Identification Sensors, Drone Fleet, and Docking Stations that are necessary for data collection in a particular sector.

Vehicles Detecting Sensor are located along the length of roads in pre-determined urban areas that are of interest for observation and are not infrastructurally covered by monitoring systems. Such areas are named as sectors. Sectors can be of different sizes and shapes. Two adjacent sectors may or may not be interconnected. Depending on the road environment, sectors may have several entrances and exits. The sensors record the number of vehicles entering the observed sector and record the number of vehicles leaving the sector in a certain time interval. The number of vehicles in each sector may indicate the need for monitoring the sector. The images from drone cams are forwarded using the Communication System to the Control Center for processing. The Perception layer must contain a hub device that collects data from sensors and drone cams and allows them to communicate with each other, as well as sends data to the Network layer.

Drone Fleet in the case of the presented IoT system refers to a group of two or more drones that perform the task of supervision as needed and belong to the same Docking Station. Drone Fleet management is performed through the Drone Management Algorithm. The drones in the fleet communicate with the Control Center and with each other. The task of the drone fleet is to monitor a specific sector. The Control Center uses the Drone Management Algorithm to manage drones in the fleet and optimize the number of drones based on hardware performance of drones, maximum flight duration and sector parameters. Also, the number of drones in the fleet depends on the number of sectors and the capacity of the Docking Station.

Docking station contains an automated system that serves drones in terms of fast battery charging, servicing, garaging and other operations related to drones. It is also used for the take-off and landing of drones, and communication with the Control Center. The proposed IoT system envisages the use of static Docking Stations, and the connection with the Control Center is established by wire (Ethernet). The Docking Station is in a place that enables safe take-off or landing and uninterrupted communication of drones.

3.2 Network Layer

The Network Layer oversees the entire communication within the presented IoT system, and with the help of the Communication System it connects the entire system. The task of the Network layer is to achieve secure data transfer (Security Layer) between the drone fleet and the Control Center with the help of the Communication System and vice versa. The management of the entire IoT system is performed in the Control Center which is equipped with a Web & Application server and a database. The Web & Application server coordinates drones, receives data from drones, stores data in a database, performs analytics, and all types of processing. It also manages the data stored in the database and manages the fleet of drones based on the processed data. Internet of Drones (IoD) is also managed by the Control Center through the Cloud-Based Drone Service Center, see Fig. 2. Web and application servers can be low performance because pre-trained neural networks can be used to process and analyze data. Also, the reduction of IoT system costs can be achieved by a strictly controlled drone fleet management system.

Communication system provides two types of communication between drones. These two ways of networking are Drone-to-Drone (D2D) and Drone-to-Infrastructure (D2I). The IoT on-demand tracking system is oriented as Drone-to-Infrastructure (D2I) as communication is established between drones and Vehicles Detecting Sensors, and drones and Docking Stations. Communication is done by the Web & Application Server located in the Control Center. The Communication System uses Bluetooth and 4G/5G technologies. Bluetooth technology is needed to establish drone communication with the Docking Station. Base stations play a key role in maintaining the drone's connection with the Control Center. 4G/5G wireless technologies enable the establishment of drone communication with base stations. Given the location of the system setup and the terrain configuration, the network infrastructure should have support for roaming on Wi-Fi networks. The TCP/IP communication protocol is in the Network Layer.

3.3 Service and Application Layer

The Service & Application Layer is based on the concept of cloud computing and contains resources and elements of artificial intelligence (AI) for data processing. The Cloud-Based Drone Service architecture is presented in further lines.

Control Center enables the management of the entire IoT system and is equipped with a Web & Application server and a Database. Web & Application server, on which Web Services are executed, coordinates drones, receives data from drones, stores data in a database, performs analytics and all kinds of processing. It also manages the data stored in the Database and manages the Drone Fleet based on the processed data. The IoD is also managed by the Control Center through the Cloud-Based Drone Service Center (see Fig. 2).

The main idea of our approach is to integrate system parts implemented by using different technologies such as Internet of Things in sensing in urban area and Drone Fleet used for on-demand monitoring problems in Smart City urban area. The Cloud-based drone service architecture is used for the integration purpose, and it is presented in Fig. 2. For the cloud-based part of the system, a Service-Oriented Architecture (SOA) was accepted since it enables building software systems as a set of independent software services that communicate to achieve certain goals [16].

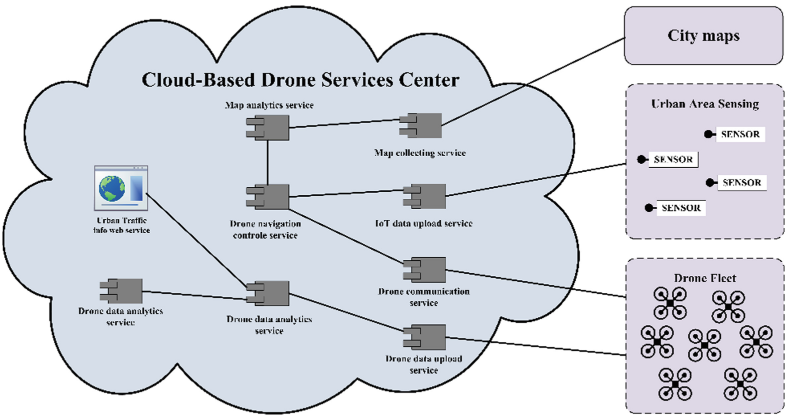


Fig. 2. Cloud-based drone service architecture

The central part is cloud-based system, named Cloud-Based Drone Services Center that is used for integration of different parts of the system. The cloud system is deployed at Linux server, while software services in the cloud are developed in JEE [17].

In the Urban Area Sensing part, sensors for counting the number of vehicles that moves between sectors in the urban area (Vehicle Detection Sensors) are deployed. These sensors send data to the cloud system, where IoT Data Upload Service collects data, do pre-processing (formatting of data) and sends data to Drone Navigation Control Service. All data necessary for further processing are stored in Database and Web & Application server.

Drone Navigation Control Service is a central software service in the cloud part of the IoT system that manages the Drone Fleet based on the Drone Management Algorithm, sending control messages via the Drone communication service. The control of drones is based on data collected from IoT sensors for counting vehicles by using IoT Data Upload Service, and data on sectors deployed on urban area map. The maps are collected by Map collecting service and delivered to Map analytics service for determining the sectors on the map.

The Service & Application Layer, with implemented Web Services, manages all applications and services in the IoT system (see Fig. 2). Drone Navigation Control Service is the central software service in the system and sends control messages via Drone communication service. The Drone Management Algorithm is executed on the Web & Application server.

Also, the Service & Application Layer has the task of integrating the data collected from drones that are uploaded with Drone data upload service, which sends data to Drone data analytics service. Based on the analysis of data obtained from drones, some info can be published by using Urban traffic info web service, while some urban emergency services can be called by using Emergency action service (police, ambulance or parking service). This part of the system will not be presented in detail in this paper. Since the focus is on controlling drone fleet, the following sections will provide details on Drone Navigation Control Service.

3.4 Management Layer

The vertical layer called Management is responsible for control and management of all the layers of the proposed IoT system (Fig. 1). Therefore, the Management Layer controls and manages the core systems in the proposed IoT architecture (Data Collection System, Communication System, Service and Application System and Security).

3.5 Security Layer

The Security Layer should cover all horizontally and vertically oriented layers in the proposed IoT concept. The IoT on-demand monitoring system includes a Drone Fleet with a low level of protection implemented. Therefore, the biggest problem in communication is the detection of anomalies in network traffic caused by DDoS (Distributed Denial of Service) attack. The security of the proposed IoT concept does not propose a new approach for the detection and elimination of network anomalies of DDoS traffic, but the use of already known security models [18].

4 Drone Fleet Operation in Urban Environment

Drone Fleet operation primarily depends on Vehicle Detector Sensors and Control Center. First, it is necessary to clearly define the concept of urban area and urban sectors within urban area (sectors in further text). The term urban area refers to a part of a Smart City or the whole Smart City in a special case. Second, sectors are defined as separate units of an urban area that do not have to be physically adjacent to each other (see Fig. 3).

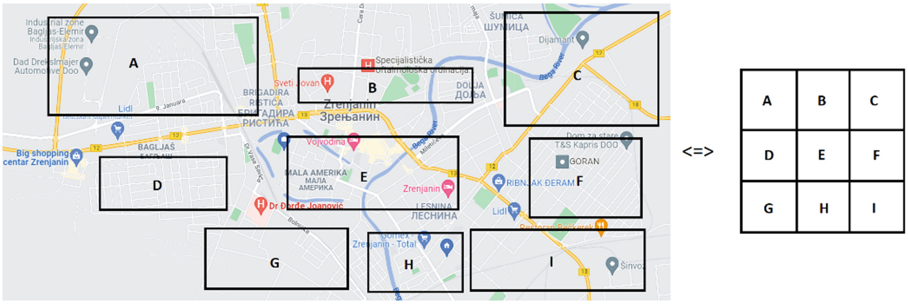


Fig. 3. Example of an urban area and sectors (3 by 3 grid)

Physical sectors are transformed to a Euclidean grid (Sector grid), and therefore the choice of sectors must enable this transformation. Also, sectors do not have to be of rectangular shape. Each of the sectors is defined by inputs (input roads) and outputs (output roads) where Vehicle Detection Sensors are located. For a time-interval [t_0, t_1] at moment t_1, it is possible to obtain the total number of vehicles that entered the sector, as well as the total number of vehicles that left the sector.

The parameters that affect the choice of urban area relate primarily to the available resources: Vehicle Detection Sensors that are pre-installed or there is a real possibility of implementing Vehicle Detection Sensors. Also, the architecture and layout (position) of Control Centers play a significant role in this. It is easy to see that the signal range (connection of the drone with the control center) is a significant parameter. In addition, the choice and size of this area, are directly related to the hardware performance of drones and maximum flight duration. The layout of battery replenishment stations (Docking Stations) also plays a significant role.

The representation of the urban area by sectors is suitable as a basis for drone management so that the sectors are further elaborated. The assumption is that sectors are accessible by air so that drones can reach them. There are several limiting factors for drone movement such as: obstacles (tall buildings…), weather conditions in general (rain, snow, wind direction and strength before all), restricted sectors, etc. Restrictive factors affect the length of the route, and thus prolong the length of the flight. Each of the sectors can contain:

- Road traffic infrastructure (intersections, roads, roundabouts, parking lots…).
- Residential infrastructure (buildings, green areas and parks…).

- Network infrastructure that allows drone operation within the sector. Possible drone functions were discussed in [1, 19] where the paths were predefined by a graph and Dijkstra [20, 21] and Floyd-Warshall [22–24] algorithms were used.

The next task of this research is to single-out the sectors that should be visited, so it is necessary to define the criteria for selecting the sector. As stated earlier, each sector is characterized by the total number of entering vehicles (summed data from all Vehicle Detection Sensors) in a certain time interval (in) and the total number of leaving vehicles (summed data from all Vehicle Detection Sensors) in the same time interval (out). The difference between inputs and outputs may be one of the indicators of the situation within the sector, but it should be borne in mind that the sectors are not the same size (area) and are different in structure. If this is considered, it would be necessary to define a threshold for each sector separately that determines whether the sector should potentially be processed. Instead, Potential Sector Load (PSL) as a criterion for sector selection is considered.

4.1 Potential Sector Load

Potential Sector Load (PSL) is a measure related to the worst-case scenario. In order to calculate PSL value, the immediate neighborhood of the sector S, 1-neighborhood, is considered, and in Fig. 4, 1-neighborhood and 2-neighborhood are shown to illustrate the concept of the n-neighborhood.

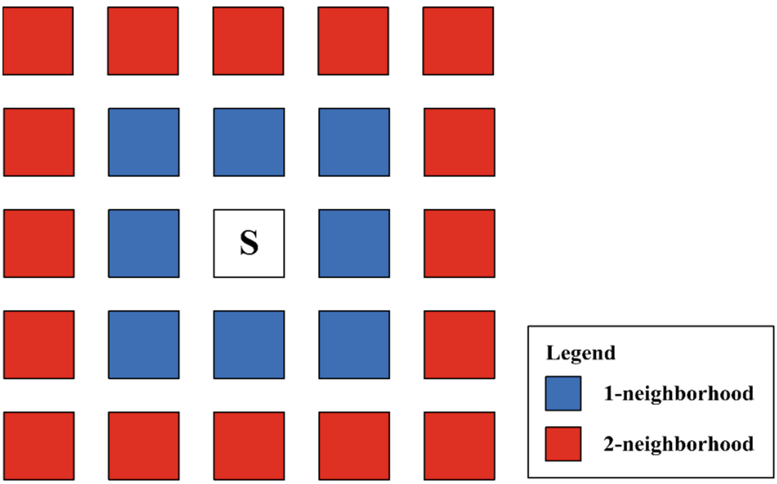


Fig. 4. Concept of n–neighborhood (1-neighborhood and 2-neighborhood)

For the sector S defined by coordinates (i, j), two values are calculated: Total entry into sector S at moment t (1) and total entry into sector S at moment t + 1 (2). Worst case scenario implies that at moment t all vehicles leaving the neighboring sectors (1-neighborhood) to S, move towards sector S, and at moment t + 1 all vehicles entering the neighboring sectors (1-neighborhood) to S, will also move towards sector S.

$$Input(t)_{i,j} = \sum_{\substack{m,\, n\, \in\, \{-1,\, 0,\, 1\} \\ m^2 + n^2 \neq 0}} out_{i+m,j+n}(t) \tag{1}$$

$$Input(t+1)_{i,j} = \sum_{\substack{m,\, n\, \in\, \{-1,\, 0,\, 1\} \\ m^2 + n^2 \neq 0}} in_{i+m,j+n}(t) \tag{2}$$

PSL value (3) for the sector S defined by coordinates (i, j) is calculated as difference between (2) and (1).

$$PSL_{i,j} = u_{i,j}(Input(t+1)_{i,j} - Input(t)_{i,j}) \tag{3}$$

In (3) u_(i, j) is sector coefficient that reflect sector features like: total number of parking spaces, usual traffic load, road quality, noise sensitivity, etc. In case when $Input(t+1)_{i,j} > Input(t)_{i,j}$ the value of $PSL_{i,j}$ is positive, meaning that there is a potential for sector overload. These calculations are executed in Control Center via *Drone navigation control service.*

4.2 Simulation

The drone management simulation was implemented in the Java programming language. Sectors are represented by a Sector grid (the obvious representation is a 2D matrix). For each sector the values of input (in) and output (out) are defined. These values consider the time of day, i.e., traffic peaks are simulated. For each sector in Sector grid the PSL value was calculated based on (3). Figure 5 shows the PSL values (Fig. 5a), and for a given threshold, the sectors in which the PSL value exceeds the specified threshold are selected (Fig. 5b).

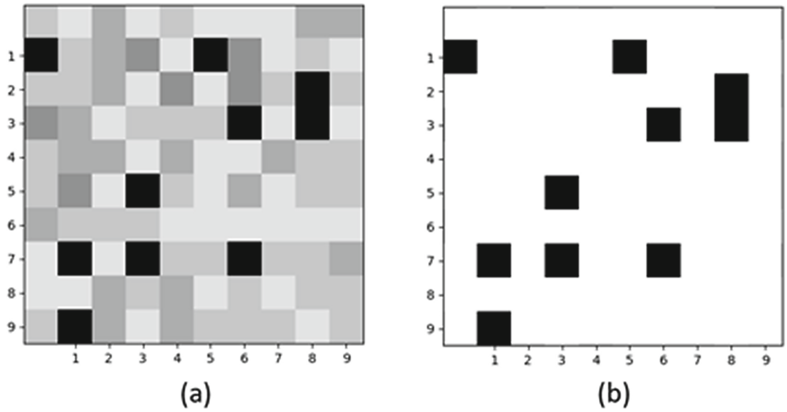


Fig. 5. Sector selection: PSLvalues (a) and complete graph nodes (b)

In this way Euclidean representation (Fig. 5a) is transformed to non-Euclidean (Fig. 5b): the selected sectors marked in black represent the nodes of the complete graph. This is a weighted complete graph where each two nodes are linearly connected by an edge having the weight equals the spatial distance. The edges of the graph are not shown for obvious reasons.

The parameters that determine the simulation are Sector grid size (Fig. 5 shows grid 10 by 10 for practical reasons only), the time of the day, the number of drones, the initial position of the drones and the threshold value.

4.3 Drone Management Procedure

The drone management procedure should determine the path for each drone. Given the low-cost variant of the system architecture, the Drone Management Algorithm must be memory and time undemanding.

The implemented procedure for drone management means that the n-neighborhood of each drone is examined, starting from 1-neigborhood. There are two obvious options for a drone current neighborhood:

1. there is not a single node whose PSL value exceeds a given threshold, and
2. there are several nodes (sectors) to visit.

In the first case, the neighborhood of the drone is expanded, after the 1-neighborhood a 2-neighborhud is considered, etc. In this way, the entire Sector grid is potentially tested. In the second case, the sector with the highest PSL value is selected. If drone priority is defined, there is no problem with two or more drones going into the same sector. The priority of drones is determined based on the remaining duration of the flight (depending on the battery) and drone location. The frequent occurrence of the drone neighborhood overlap problem also depends on the size of the Sector grid, number of drones and the initial positions of the drones.

For each drone, the n-neighborhood concept allows the application of multiple heuristic search strategies. In this case, the heuristic can be defined as follows: the most critical of the closest sectors has the advantage. In other words, sectors are ranked first by distance and then by criticality. Drone management procedure is executed in Web & Application server via Drone navigation control service.

A simulation of the proposed solutions was performed, and the discussion is presented in the following section.

5 Results and Discussion

The conducted simulation examines the influence of the number of drones and the dimensions of the Sector grid on the cost and the number of iterations required to visit all selected sectors. The cost value is calculated based on the total number of sectors that the drones visited (including those sectors that drones have just flown over), so it is only a rough indication of the total path cost that the drones would achieve in real-life circumstances. The number of iterations required for a Drone Fleet to visit all selected sectors is a rough indication of the time required. PSL values were calculated for each sector and normalized to [0, 255], while threshold was set to 150.

Two scenarios were simulated:

1. Sector grid size is constant while drone count variate.
2. Number of drones is constant while Sector grid size variate.

For each of the two scenarios, the simulation was repeated 1000 times, with the time during the day being constant (08:00 am), sector coefficient was set to 1 for all sectors and the displayed values of cost and number of iterations being mean values. Results are presented by tables and graphically. Table 1 shows the variation in cost and number of iterations depending on the number of drones in the fleet. Sector grid size is set to 50 for both dimensions.

Table 1. Cost and number of iterations depending on the number of drones.

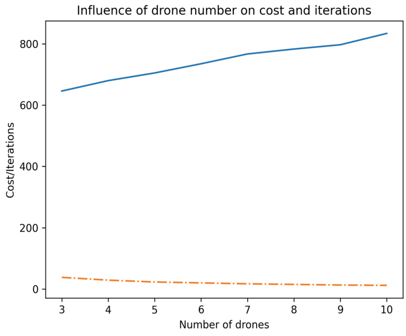
Drones:	3	4	5	6	7	8	9	10
Cost	646	680	705	735	767	783	797	834
Iterations	38	29	23	20	17	15	13	12

Table 2 shows the variations in cost and number of iterations for a constant number of drones (five), with the Sector grid size varying in both dimensions.

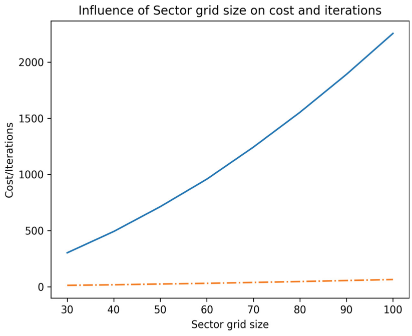
Table 2. Cost and number of iterations depending on the sector grid size.

Sector grid size:	30	40	50	60	70	80	90	100
Cost	302	492	713	958	1243	1553	1891	2256
Iterations	12	17	24	30	38	46	55	64

Graphical representation is given in Fig. 6.



(a)



(b)

Fig. 6. Total cost and number of iterations influenced by the number of drones (a) and Sector grid size (b) (Color figure online)

The blue line indicates the cost, while the orange dashed line indicates the number of iterations. Both simulation scenarios confirm that the chosen heuristics for drone management work as expected. Table 1 and Fig. 6a show that increasing the number of drones in the fleet increases the total cost but decreases the number of iterations (time). On the other hand, the increase in the dimensions of the Sector Grid, with a constant number of drones, significantly affects the increase in cost, while the number of iterations is slightly growing (see Table 2 and Fig. 6b).

The problem that is becoming obvious is determining the optimal number of drones. This is a complex problem that depends on several parameters and is not considered in this paper.

In addition, two issues are interesting to consider. Namely, there is a possibility of changing the PSL value of the sector during the drone movement. In the case of a decrease in the PSL value below the preset threshold, the sector should not be selected, i.e., it is no longer active, which results in the removal of node and edges of the complete graph. The proposed method of drone management is not sensitive to these changes, which is certainly an advantage.

A problem that is also becoming apparent is the involvement of a Drone Fleet in the case of a small number of sectors whose PSL value exceeds the predefined threshold. This is the case of single sector with a PSL value greater than threshold, or there are several "scattered" sectors of this type. Such a situation corresponds to some night traffic regimes or special situations. Figure 7 shows examples of several situations. Figure 7a shows a common case of Sector grid to be processed. Figure 7b shows the case of a smaller number of active sectors but they are grouped, which may be the reason for drone fleet engagement. Figures 7d, 7e, and 7f show examples of the Sector grid that may not require Drone Fleet engagement.

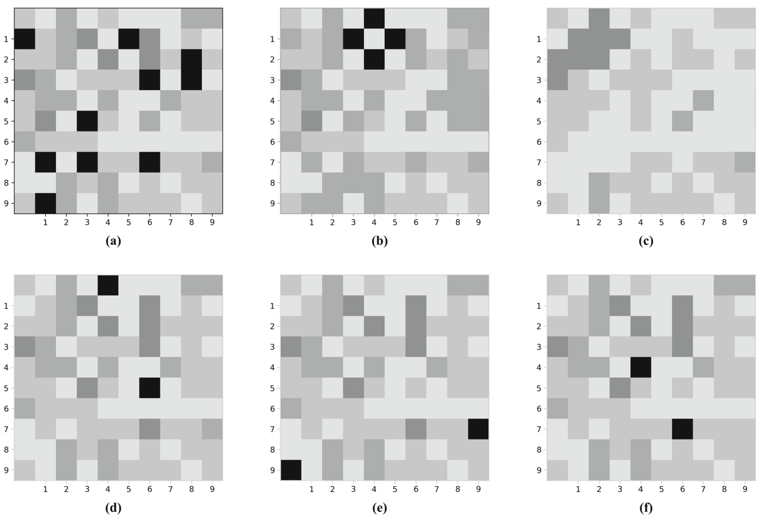


Fig. 7. Sector grids that need to be processed (a), (b), (c) and Sector grids that do not need to be processed (d), (e), (f)

The case that stands out is shown in Fig. 7c, there are no active sectors but there is a group of sectors that are congested with traffic, but no sector alone exceeds the PSL threshold. This situation may also be the reason for the drone fleet engagement. A situation with a small number of close sectors whose PSL values are higher or close to the threshold of m × n dimension Sector grid, is easily recognized by creating a frame (window) of dimensions i × j where i < m and j < n that would slide across the Sector grid and check the PSL of sectors it covers. However, it is possible to use Convolutional Neural Networks (CNN) to identify such potentially critical sectors. Based on previous experience, proposed CNN architecture is:

1. Convolution2D (kernel = 128, kernel_size = (3, 3), activation = 'relu')
2. MaxPooling2D (pool_size = (3, 3))
3. Dense (64, activation = 'relu')
4. Dense (1, activation = 'sigmoid')

A trained CNN can recognize a Sector grid that contains critical sectors, even if none of the sectors exceeds the PSL threshold. This solution enables generalization, i.e., identification of potentially critical situations that have not been seen before.

6 Conclusions

This paper deals with the management of the Drone Fleet in the areas of Smart Cities that are not infrastructural covered by continuous monitoring systems. Attention was primarily focused on the problem of drone control and management, with the aim of determining the locations that drones should visit. The proposed approach involves the use of a Drone Fleet working over sectors of the urban area.

First, the proposed IoT architecture is cloud-based and implies the existence of a Vehicle Detection Sensors. The basic idea was to integrate parts of the system implemented using different technologies such as the Internet of Things and a Drone Fleet. On-demand IoT monitoring system as well as drone management using Web Services are described.

Second, the sector grid generation procedure and the procedure for selection of critical sectors were proposed. These procedures are based on n-neighborhood concept and tend to be general and suitable for implementation. The Drone Management Algorithm also uses the concept of n-neighborhood and is not sensitive to the dynamic nature of traffic, given that the set of critical sectors is changing. The simulation conducted on two scenarios shows the impact of drone number and sector grid size on the path cost and the total time required to visit critical sectors. The criticality of the sector depends on the number of vehicles entering/leaving the sector. The results of the simulation indicate that increasing the number of drones increases the total path cost, but reduces the time required to visit all critical sectors. Increasing the size of the urban area also increases the total path cost and time required.

The possible roles of drones within the sector have not been considered. However, the role within the sector can be simple recording the situation, monitoring of traffic load, calculation and recalculation of routes presented by the graph, as well as monitoring of improperly parked vehicles. The number of possible roles within the sector is certainly higher.

It is useful to look at the prospects of real-life implementation. The research described in other papers provides insight into the problems of Drone Fleet management and parameters optimization. The possibility of using Graph Neural Networks (GNN), whose training requires significant resources, should be emphasized here, although the use of trained GNN is significantly less resource intensive. In addition, the choice of drone starting points and battery recharge station (Docking Station) positions is becoming clear, which is a separate study. Given that low-cost implementation in real-life conditions was anticipated, the assumption is that process power is limited. Also, the content of previous work provides a clear insight into the impact of weather conditions, which can be a limiting factor for implementation in real life. Therefore, the proposed solution allows the formation of nodes and an efficient and easy to implement Drone Management Algorithm.

The advantages of the IoT on-demand monitoring system are scalability and decentralization. The scalability of the IoT system can be achieved by developing Client Applications that are directed towards citizens and police, ambulance, and parking service. This is partially considered in the cloud-based architecture as Emergency action service, (see Fig. 2). Decentralization of the system can be done by each Docking Station taking the role of Control Center.

Future research may relate to the improvement of the Drone Management Algorithm as well as the further use of convolutional neural networks, primarily for the assessment of critical situations in Smart City traffic. The method of sector selection and the selection of significant sector parameters are also possible directions for future research.

Acknowledgments. Ministry of Education, Science and Technological Development, Republic of Serbia financially supported this research, under the project number TR32044: "The development of software tools for business process analysis and improvement", 2011–2021.

References

1. Jausevac, G., Dobrilovic, D., Brtka, V., Jotanovic, G., Perakovic, D., Stojanov, Z.: Smart UAV monitoring system for parking supervision. In: Perakovic, D., Knapcikova, L. (eds.) FABULOUS 2021. LNICSSITE, vol. 382, pp. 240–253. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-78459-1_18
2. Dobrilović, D., Brtka, V., Jotanović, G., Stojanov, Ž, Jauševac, G., Malić, M.: Architecture of IoT system for smart monitoring and management of traffic noise. In: Knapčíková, L., Peraković, D., Behúnová, A., Periša, M. (eds.) 5th EAI International Conference on Management of Manufacturing Systems. EICC, pp. 251–266. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-67241-6_21
3. Koubâa, A., et al.: Dronemap planner: a service-oriented cloud-based management system for the internet-of-drones. Ad Hoc Netw. **86**, 46–62 (2019)
4. Gubbi, J., Buyya, R., Marusic, S., Palaniswami, M.: Internet of Things (IoT): a vision, architectural elements, and future directions. Futur. Gener. Comput. Syst. **29**, 1645–1660 (2013). https://doi.org/10.1016/j.future.2013.01.010

5. Koubâa, A., Qureshi, B.: DroneTrack: cloud-based real-time object tracking using unmanned aerial vehicles over the internet. IEEE Access **6**, 13810–13824 (2018)
6. Capello, E., Dentis, M., Guglieri, G., Mascarello, L.N., Cuomo, L.S.: An innovative cloud-based supervision system for the integration of RPAS in urban environments. Transp. Res. Procedia **28**, 191–200 (2017)
7. Hu, L., et al.: "CloudStation:" a cloud-based ground control station for drones. IEEE J. Miniaturization Air Space Syst. **2**, 36–42 (2020)
8. Mehrooz, G., Ebeid, E., Schneider-Kamp, P.: System design of an open-source cloud-based framework for internet of drones application. Presented at the 2019 22nd Euromicro Conference on Digital System Design (DSD) (2019)
9. Peng, H., et al.: Dynamic graph convolutional network for long-term traffic flow prediction with reinforcement learning. Inf. Sci. **578**, 401–416 (2021)
10. Pei, Z., Dai, X., Yuan, Y., Du, R., Liu, C.: Managing price and fleet size for courier service with shared drones. Omega **104**, 102482 (2021)
11. Ghelichi, Z., Gentili, M., Mirchandani, P.B.: Logistics for a fleet of drones for medical item delivery: a case study for Louisville, KY. Comput. Oper. Res. **135**, 105443 (2021)
12. Faraci, G., Raciti, A., Rizzo, S.A., Schembra, G.: Green wireless power transfer system for a drone fleet managed by reinforcement learning in smart industry. Appl. Energy **259**, 114204 (2020)
13. Li, J., Sun, P., Hu, Y.: Traffic modeling and optimization in datacenters with graph neural network. Comput. Netw. **181**, 107528 (2020)
14. Kapoor, A.: Hands-On Artificial Intelligence for IoT: Expert Machine Learning and Deep Learning Techniques for Developing Smarter IoT Systems. Packt Publishing (2019)
15. Jotanovic, G., Brtka, V., Curguz, Z., Stojcic, M., Eremija, M.: Mobile applications for recording road traffic noise. In: Proceedings 8th International Conference on Applied Internet and Information Technologies, "St Kliment Ohridski" University-Bitola, Faculty of Information and Communication Technologies-Bitola, Bitola, Republic of Macedonia, pp. 94–98 (2018)
16. Rotem-Gal-Oz, A.: SOA Patterns. Simon and Schuster (2012)
17. Kumar, B.V.: Implementing SOA Using Java EE. Pearson Education India (2010)
18. Cvitić, I., Peraković, D., Periša, M., Botica, M.: Novel approach for detection of IoT generated DDoS traffic. Wirel. Netw. **27**(3), 1573–1586 (2019). https://doi.org/10.1007/s11276-019-02043-1
19. Dobrilović, D., Brtka, V., Jotanović, G., Stojanov, Ž, Jauševac, G., Malić, M.: The urban traffic noise monitoring system based on LoRaWAN technology. Wirel. Netw. **28**(1), 441–458 (2021). https://doi.org/10.1007/s11276-021-02586-2
20. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numer. Math. **1**, 269–271 (1959)
21. Junior, D.P., Wille, E.C.G.: FB-APSP: a new efficient algorithm for computing all-pairs shortest-paths. J. Netw. Comput. Appl. **121**, 33–43 (2018)
22. Floyd, R.W.: Algorithm 97: shortest path. Commun. ACM **5**, 345 (1962)
23. Anderson, J.: Discrete Mathematics with Combinatorics Pearson (2004)
24. Aini, A., Salehipour, A.: Speeding up the Floyd-Warshall algorithm for the cycled shortest path problem. Appl. Math. Lett. **25**, 1–5 (2012). https://doi.org/10.1016/j.aml.2011.06.008