UDC 004.738.5:004.7: 621.39 doi: https://doi.org/10.20998/2522-9052.2025.4.09

Oleksandr Shefer, Stanislav Myhal

National University "Yuri Kondratyuk Poltava Polytechnic", Poltava, Ukraine

A METHOD FOR DEVELOPING SUPPORT NETWORKS FOR HIGH-DENSITY INTERNET OF THINGS THROUGH THE INTEGRATION OF SDN AND MEC TECHNOLOGIES

Abstract. Background. The deployment of high- and ultra-high-density Internet of Things (IoT) systems poses a number of technical and organizational challenges. One promising approach to addressing these challenges is the integration of advanced network and data processing technologies, in particular Software-Defined Networking (SDN) and Multi-Access Edge Computing (MEC). **This paper aims** to explore the process of integrating SDN and MEC technologies into support infrastructures for high- and ultra-high-density IoT. **Results.** The study proposes an architecture for an integrated IoT–SDN–MEC system comprising both terrestrial and aerial segments. A mathematical model has been developed for this architecture, enabling the evaluation of energy consumption in fog-layer devices as well as the estimation of task execution delays. In addition, a traffic offloading scheme for the integrated IoT–MEC–SDN system is presented. The research formulates the problem of optimizing energy consumption and task processing delays in the aerial segment. To address this problem, the Grey Wolf Optimizer (GWO) algorithm is employed, providing efficient near-optimal solutions. **Conclusion.** Simulation results demonstrate that incorporating fog-layer resources within the aerial segment of the **integrated IoT–MEC–SDN system significantly re**duces both average energy consumption and average task processing delays in high- and ultra-high-density IoT environments. **Future research** will focus on determining the optimal structural configuration of the aerial segment.

Keywords: Internet of Things (IoT); Software-Defined Networking (SDN); Multi-Access Edge Computing (MEC); high-density; ultra-high-density; integrated system; terrestrial segment; aerial segment.

Introduction

The Internet of Things (IoT) is one of the most rapidly growing technologies of our time [1]. Its popularity is driven by a wide range of applications, spanning from personal wearable devices to complex industrial systems [2, 3]. The deployment of IoT enhances business efficiency, reduces resource consumption, and enables the collection and analysis of massive data streams in real time [4]. The growth of wireless technologies and cloud platforms has significantly contributed to the widespread adoption of IoT [5]. Analysts predict that the number of connected devices will continue to increase exponentially in the coming years. By 2026, the number of IoT devices is expected to exceed 25 billion, and by the end of the decade, more than 40 billion [6]. Thus, IoT is gradually evolving from an innovative trend into a key pillar of the digital economy. One of the critical directions in IoT development is high-density IoT (HD-IoT) [7] and ultra-dense IoT (UD-IoT) [8], environments where a massive number of connected devices operate within a limited area. Such systems require spectrum optimization, ultra-low latency, and the ability to support thousands of simultaneous connections. Ultra-dense IoT deployments may involve tens or even hundreds of thousands of devices per square kilometer [9].

However, the implementation of high- and ultra-dense IoT gives rise to a number of technical and organizational challenges, including [10, 11]:

– Network congestion, where large volumes of simultaneous connections cause packet loss, excessive delays, and channel instability;

– Spectrum limitations, as frequency resources are not always sufficient to ensure stable operation for thousands of devices within a small area;

– Energy consumption, since sensors and devices in dense environments require energy-efficient protocols, otherwise leading to increased maintenance costs and frequent battery replacements;

– Scalability and management, as operating a network with tens of thousands of nodes necessitates new automation methods.

A wide range of scientific studies have sought to address these challenges. For example, [12] proposes decomposing aggregate data flows directed to cloud data centers. While this approach can reduce latency compared to traditional methods, it does not account for the specifics of the aerial segment, particularly energy efficiency. The issue of energy efficiency in aerial components of IoT support systems has been partially addressed in [13], though without considering the requirements of high-density IoT. Certain aspects of high-density IoT are discussed in [14], but the possibility of traffic offloading to the aerial segment is not explored. In [15], the focus is on improving data transmission rates, but similar to the approach in [16], this comes at the expense of higher energy consumption.

One promising approach to overcoming these issues is the integration of advanced networking and data processing technologies, particularly Software-Defined Networking (SDN) and Multi-Access Edge Computing (MEC) [17, 18]. Accordingly, **the purpose** of this paper is to investigate the integration of SDN and MEC technologies into support systems for high- and ultra-dense IoT.

1. The Potential for Integrating SDN and MEC Technologies

Multi-access Edge Computing (MEC) is an edge computing paradigm that enables data processing at the network periphery [19]. In addition to edge-level deployments, MEC can also be extended into the fog layer, thereby providing a broader distributed infrastructure [20]. The core idea of MEC is to shift data processing closer to the end user—at the “edge” of the network (e.g., at mobile operator base stations or local servers)—instead of transmitting data to remote cloud data centers. This approach offers several advantages [21]:

– reduced latency;

– faster responsiveness for latency-critical applications (e.g., AR/VR, autonomous vehicles, smart factories);

– offloading of the transport network;

– computational resources and services are brought closer to end users.

Software-Defined Networking (SDN) is a network management concept that decouples the control plane from the data plane, transferring network control from hardware devices (routers, switches) to a centralized controller that defines the operational rules of the entire network [22]. This approach allows [23]:

– flexible management of data flows;

– optimization of resource allocation;

– simplification of automation and scalability processes;

– rapid deployment of new services;

– logical, software-based network control.

The fundamental distinction between these technologies lies in their focus: **MEC** addresses the *location* of data processing (bringing computation closer to the user), while SDN addresses the *method* of network control (centralized, software-defined traffic management). Importantly, the two technologies can complement one another in complex networking scenarios [24]. MEC requires fast and flexible routing management, which is effectively supported by SDN. Conversely, SDN gains new relevance in optimizing traffic flows toward MEC-enabled edge and fog devices.

One of the most significant application domains for such integration is the support of high-density and ultra-dense Internet of Things (IoT) environments.

2. Architecture of the Integrated IoT–SDN–MEC System

The proposed integrated system consists of terrestrial and aerial segments, as illustrated in Fig. 1.

In the proposed model, the terrestrial segment represents a high- or ultra-high-density IoT network comprising distributed end devices. This is a three-layer network, whose operation relies on distributed edge computing technologies [25].

The first layer includes distributed IoT devices such as sensors and actuators. These devices can, for example, measure environmental parameters. Under high- and ultra-dense deployment scenarios, their number becomes extremely large, with wide geographic distribution and support for diverse interfaces. The second layer consists of distributed edge-computing devices within the terrestrial segment. The third layer is formed by IoT gateways, which serve as the interface between terrestrial edge devices and aerial fog servers. It is assumed that all IoT gateways in the considered network are connected to the aerial plane [26].

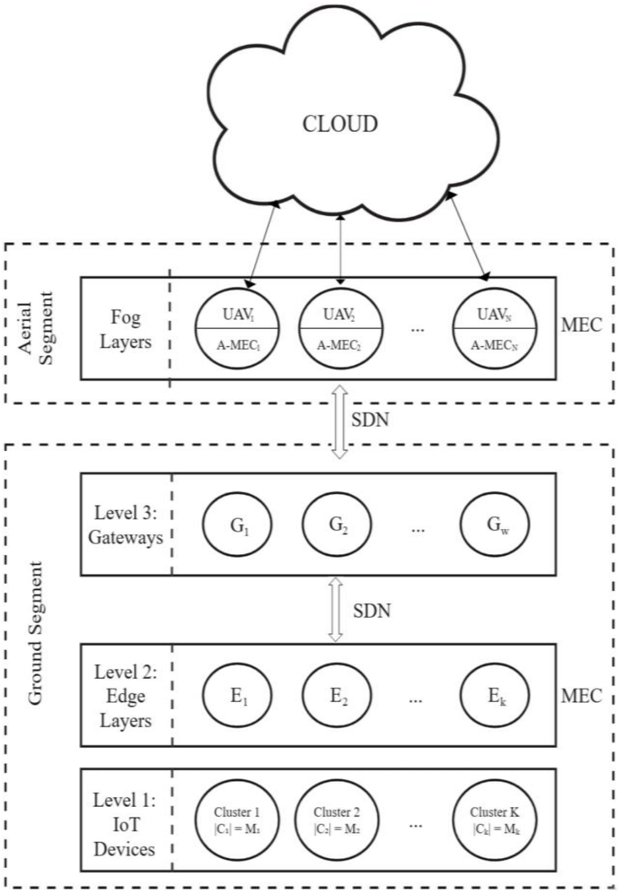

Fig. 1. Model of interaction within the integrated IoT–SDN–MEC system

The aerial segment of the model comprises multiple unmanned aerial vehicles (UAVs)—such as microdrones and quadcopters—deployed to support high- and ultra-high-density IoT networks. Each UAV is equipped with two communication interfaces: air-to-ground (A2G) and air-to-air (A2A). The A2G interface provides communication with the terrestrial segment, while the A2A interface supports communication between UAVs [27, 28].

Furthermore, each UAV hosts an Air MEC (A-MEC) server, which provides computational resources to the terrestrial network [29]. The A-MEC operates as a micro-cloud server enabling traffic offloading. Any UAV can be allocated to serve a specific cluster of IoT devices via IoT gateways, which connect the corresponding edge device to one of the UAVs using SDN technology.

3. Mathematical Model of the Integrated IoT–MEC–SDN System

In the aerial segment of the system, a swarm of unmanned aerial vehicles (UAVs) is deployed. Each UAV is equipped with an Air MEC (A-MEC) server capable of both processing information locally and transmitting it to a remote cloud data center. This UAV swarm is described by the set D:

$$D = \{D_1, D_2, \dots D_N\}, \quad (1)$$

where N is the total number of UAVs deployed in the aerial plane, and D_i is an element of set D corresponding to the UAV with index i, $i \in \overline{1, N}$.

A three-dimensional Cartesian coordinate system with coordinates (x, y, h) is used to determine the positions of UAVs and IoT devices. The position of a UAV is characterized by two-dimensional ground coordinates (x, y) and altitude h. Thus, the current location of the UAV D_i at time t is given by the coordinates $(x_i(t), y_i(t), h_i)$. IoT end devices are distributed across the plane (X, Y). Each UAV follows a trajectory $A_i(t) = (x_i(t), y_i(t), h_i)$ at a fixed altitude h_i. The network of IoT end devices is described by the set I:

$$I = \{I_1, I_2, \dots I_M\}, \quad (2)$$

where M is the total number of IoT devices, and I_j is an element of set D, corresponding to the IoT device associated with the UAV of index j, $j \in \overline{1, M}$.

The position of each IoT device on the plane (X, Y) is characterized by two-dimensional coordinates (x, y). Thus, the current position of the j-th IoT device is specified by the coordinates (x_j, y_j).

IoT devices are grouped into clusters. Within each cluster, all devices transmit data packets to a fixed IoT edge server, which determines the subsequent processing actions: either processing the information locally at the edge or forwarding it to the nearest available IoT gateway [30].

The set of deployed IoT clusters is described by the set C:

$$C = \{C_1, C_2, \dots, C_K\}, \quad (3)$$

where K is the total number of IoT edge servers, and correspondingly the total number of IoT clusters; C_k is an element of set C representing the cluster with index k, $k \in \overline{1, K}$,

$$C_k = (SG_k, CI_k), \quad (4)$$

where SG_k denotes the IoT edge server that forms the cluster with index k; CI_k is the set of IoT devices belonging to cluster k,

$$CI_k = \{C_{k1}, C_{k2}, \dots, C_{kM_k}\}, \quad CI_k \subset I, \quad (5)$$

where the elements of this set are the indices of IoT devices from set I; M_k is the number of IoT devices in cluster k, such that:

$$\sum_{k=1}^{K} M_k = M. \quad (6)$$

The set of IoT gateways is defined as G:

$$G = \{G_1, G_2, \dots, G_W\}, \quad (7)$$

where W is the total number of IoT gateways, G_w is an element of set G, corresponding to the gateway with index w, $w \in \overline{1, W}$, which has coordinates (x_w, y_w).

The distance between the i-th UAV and the w-th gateway at time t is given by:

$$\ell_{iw}(t) = \sqrt{(x_i(t) - x_w)^2 + (y_i(t) - y_w)^2 + h_i(t)}. \quad (8)$$

Each i-th UAV has a limited flight duration L_i, which depends on the energy consumed by computation, communication, and flight processes. The processes running on one UAV differ from those on another, and therefore the flight durations of different UAVs are not the same.

The total flight time is divided into time slots of fixed duration Δt. This allows the continuous process under consideration to be modeled as a discrete one. Within this model, both communication and computation processes are executed in these slots. During each time slot Δt, the position of each UAV is approximated as fixed. The set of such positions corresponds to the UAV trajectory and can be defined through the following finite sequence:

$$B_i(\Delta t) = \begin{pmatrix} (x_i(\Delta t), y_i(\Delta t)), \\ (x_i(2 \cdot \Delta t), y_i(2 \cdot \Delta t)), \dots, \\ (x_i(b_i \cdot \Delta t), y_i(b_i \cdot \Delta t)) \end{pmatrix}, \quad (9)$$

where denotes the time slot b_i in which the energy resources of the i-th UAV are depleted.

The energy consumed by computation is calculated based on the amount of data offloaded. The energy consumed by communication depends on the volume of data transmitted via the inbound links from IoT gateways and the outbound links delivering information to the cloud data center. The total energy consumption at the beginning of the ξ-th time slot is computed as:

$$\begin{gathered} \Omega_i(\xi \cdot \Delta t) = \Omega_{calc\_i}(\xi \cdot \Delta t) + \\ + \Omega_{connect\_i}(\xi \cdot \Delta t) + \Omega_{flight\_i}(\xi \cdot \Delta t), \end{gathered} \quad (10)$$

where $\Omega_{calc\_i}(\xi), \Omega_{connect\_i}(\xi), \Omega_{flight\_i}(\xi)$ – represents the energy consumption of the i-th UAV at the beginning of the ξ-th time slot due to computation, communication, and flight processes, respectively.

In ultra-dense IoT networks, to increase the number of devices that can be served within a single time-frequency resource while reducing latency and energy consumption, Non-Orthogonal Multiple Access (NOMA) is applied. With NOMA, all end devices use the same frame duration t_s for data transmission simultaneously [31]. Therefore, for UAVs, it is reasonable to select the discretization time slot equal to the NOMA frame duration, i.e., $\Delta t = t_s$.

4. Traffic Offloading in Integrated IoT–MEC–SDN Environments

This section introduces a method for traffic offloading from terrestrial IoT infrastructures to mobile fog computing servers deployed on unmanned aerial vehicles (UAVs). At the edge layer, IoT devices utilize their limited on-board resources to process data locally. Such an approach is viable only for lightweight tasks with modest computational requirements. Tasks that demand more intensive resources, however, must be offloaded to fog servers in the aerial segment or to remote cloud data centers [32]. To simplify the model, we adopt a binary offloading strategy. UAVs are assumed to be dedicated exclusively to supporting high-density and ultra-dense IoT networks, without other local workloads. In this context, UAVs operate as mobile fog servers, capable of directly processing tasks offloaded by IoT devices or redirecting them to other computational nodes. Traffic exchange between terrestrial and aerial layers relies on Non-Orthogonal Multiple Access (NOMA), which enhances connectivity and spectral efficiency across device clusters.

IoT traffic offloading operates on three hierarchical levels. At the edge, SDN-enabled devices employ a software profiler to determine task specifications, including the data volume (in bytes) and the required number of CPU cycles for processing a single data unit. A resource scheduler then provides the SDN decision engine with real-time information on available resources, estimates processing time and energy consumption, and calculates the residual energy after task execution.

Based on these estimates of quality of service (QoS), latency, and energy efficiency, the decision engine determines whether the task should be executed locally or forwarded via a gateway to the MEC fog layer. If sufficient resources and energy are available, the system avoids unnecessary offloading. Conversely, when resources or energy are insufficient, the edge device initiates offloading either to a neighboring device (if available) or to a fog server via the gateway.

At the fog layer, the IoT gateway issues an offloading request to the aerial MEC infrastructure. Each request contains essential task parameters along with device identification. The MEC system processes these requests and consults its decision-making module, which evaluates the availability of computing resources in the aerial segment and the QoS constraints of the task.

The A-MEC edge server of the selected UAV processes the received offloading requests from serviced end devices by extracting device and task information and forwarding them to the A-MEC decision-making mechanism. The decision-making mechanism also receives information about currently available resources from the resource scheduler and calculates the time required to process the requested task. Then, a decision is made to accept or reject the offloading request to this UAV by comparing the total required processing time with the time limits necessary to ensure QoS. If the decision is negative, the UAV requests resources from a neighboring UAV. Offloading requests are thus transferred among UAVs within the swarm.

The binary decision on accepting an offloading request is determined by comparing the residual UAV energy after completing the requested task with the UAV's energy threshold. The UAV first calculates the energy cost for task execution and the energy required for transmitting the computed results. If the UAV refuses to accept the offloading request, it forwards the request to another UAV in the swarm with available resources. Neighboring nodes respond with acceptance or rejection of the offloading requests, and the UAV either finalizes the task or offloads it based on the received responses.

5. Optimization of Energy Consumption and Task Processing Delay in the Aerial Segment

The considered offloading approach, in terms of both energy consumption and delay associated with executing IoT computational tasks in the aerial fog segment, is largely determined by the positioning of UAVs.

The primary objective of this subsection is to jointly optimize energy consumption and minimize the average latency for executing offloaded IoT tasks. This formulation leads to a two-parameter nonlinear programming optimization problem with significant computational complexity.

To reduce the complexity of the problem, we introduce weighting coefficients that balance the relative importance of temporal and energy characteristics, denoted as β_{time} та β_{energy}, respectively:

$$\beta_{time} + \beta_{energy} = 1. \quad (11)$$

Thus, if $\beta_{time} = 0$, optimization is performed exclusively with respect to energy consumption; conversely, if $\beta_{energy} = 0$ the optimization criterion depends solely on latency. For any other values of the weighting coefficients, the optimization problem is reduced to a single-parameter task of determining the optimal distribution of IoT tasks among UAVs in the aerial fog segment.

Consider a single operational cycle of a dense IoT system with duration A time slots. During each slot ξ $\left(\xi \in \overline{1, A}\right)$ tasks R_ξ are offloaded from the IoT edge layer to the aerial segment.

Boolean variables $\delta_{\xi r_\xi i}$ are defined such that $\delta_{\xi r_\xi i} = 1$ if, at slot α task r_ξ $\left(r_\xi \in \overline{1, R_\xi}\right)$ is executed by UAV i, and $\delta_{\xi r_\xi i} = 0$ otherwise. Based on this definition, the set of possible allocations can be represented as

$$\gamma = \left\{\delta_{\xi r_\xi i}\right\}. \quad (12)$$

The objective function of the optimization problem can be formulated as

$$\sum_{\xi=1}^{A} \sum_{r_\xi=1}^{R_\xi} \sum_{i=1}^{N} \left(\begin{array}{c} \beta_{time}\left(\delta_{\xi r_\xi i} \cdot T_{\xi r_\xi i}\right) + \\ +\beta_{energy} \cdot \delta_{\xi r_\xi i} \times \\ \times\left(\Omega_{calc\_\xi r_\xi i} + \Omega_{connect\_\xi r_\xi i}\right) \end{array} \right) \xrightarrow{\gamma} \min, \quad (13)$$

where $T_{\xi r_\xi i}$, $\Omega_{calc\_\xi r_\xi i}$ and $\Omega_{connect\_\xi r_\xi i}$ denote the temporal and energy costs of computation and communication, respectively, when task r_ξ at a slot ξ is processed by i-th UAV.

A set of constraints must also be introduced to ensure feasibility.

The first constraint guarantees that the velocity of any UAV does not exceed the maximum permissible swarm velocity V_{max}, i.e., at each slot for every UAV the following condition holds [33]:

$$\sqrt{\begin{pmatrix} \left(x_i\left(\xi\cdot\Delta t\right)-x_i\left(\xi\cdot\Delta t+\Delta t\right)\right)^2+ \\ +\left(y_i\left(\xi\cdot\Delta t\right)-y_i\left(\xi\cdot\Delta t+\Delta t\right)\right)^2 \end{pmatrix}}\le V_{max} \qquad (14)$$
$$\forall i\in\overline{1,N},\ \forall\xi\in\overline{1,A}.$$

The next two constraints ensure that each UAV covers only its designated flight region and that collisions among UAVs within the swarm are avoided:

$$x_{i\_\min}<x_i\left(t\right)<x_{i\_\max}$$
$$\forall i\in\overline{1,N},\ \forall t\in\overline{0,A\left(\Delta t+1\right)}; \qquad (15)$$

$$y_{i\_\min}<y_i\left(t\right)<y_{i\_\max}$$
$$\forall i\in\overline{1,N},\ \forall t\in\overline{0,A\left(\Delta t+1\right)}, \qquad (16)$$

where $x_{i\_\min}, x_{i\_\max}, y_{i\_\min}, y_{i\_\max}$ represent the minimum and maximum admissible coordinates of i-th UAV in the deployment plane.

Another constraint enforces the binary nature of offloading decisions:

$$\delta_{\xi r_\xi i}\in\left\{0;1\right\}\quad \forall i\in\overline{1,N},\ \forall\xi\in\overline{1,A},\ \forall r_\xi\in\overline{1,M}. \qquad (17)$$

The final constraint ensures that the total number of offloading requests directed to the aerial segment does not exceed the number of participants in the cluster:

$$R_\xi\le N\quad \forall\xi\in\overline{1,A}. \qquad (18)$$

Expressions (13)–(18) thus define a large-scale nonlinear programming problem with Boolean variables, addressing the optimization of IoT task allocation in the aerial fog computing segment.

6. Metaheuristic Algorithm for Approximate Search of Optimal Positions of Aerial Fog Devices

Due to the high dimensionality of the optimization problem, an approximate population-based algorithm was selected for its solution. The following metaheuristic algorithms were considered: the Grey Wolf Optimizer (GWO), the Particle Swarm Optimization algorithm (PSO), and the Salp Swarm Algorithm (SSA).

All three employ iterative search processes and are inspired by biological models, but they differ in their mechanisms of population updating and in the evolution of their topological structures [34–36].

In general, metaheuristic algorithms such as GWO, PSO, and SSA are powerful and efficient tools applicable to a broad class of optimization problems.

The choice of algorithm depends on the characteristics of the task and the performance requirements. For instance, PSO is more resource-intensive than the other two algorithms, which affects its convergence speed and computational time.

After analyzing the specific features of these algorithms, particularly regarding convergence rate and parameter tuning simplicity, the Grey Wolf Optimizer (GWO) was selected. GWO is a metaheuristic inspired by the social hierarchy and hunting strategy of grey wolves, which live and act in coordinated packs led by an alpha wolf. The algorithm initializes a population of wolves (candidate solutions) and evaluates their fitness values. At each iteration, wolves update their positions based on both their own current state and the relative positions of other wolves in the pack. As a result, at every iteration, including the final one, the algorithm produces from one to four ordered solutions corresponding to UAVs suitable for subsequent task offloading from the gateway. Each grey wolf in the pack represents a point in the solution search space with a computed fitness function value. According to fitness quality, the pack is divided into four categories. Three wolves with the best fitness values are assigned leadership roles: α, β, and δ. The remaining wolves form the category ω. The position of each wolf at iteration $t+1$, which reflects the encirclement of prey, is updated according to the following expression:

$$X_i\left(t+1\right)=X_{p,i}-$$
$$-A_i\cdot\left|C_i\cdot X_{p,i}-X_i\left(t\right)\right|,\quad i=\overline{1,N}, \qquad (19)$$

where t is the iteration index, i denotes the component of the N-dimensional space, $\vec{X}\left(t\right)$ is the position vector of the current wolf, $\vec{X}_p$ is the position vector of the prey (representing the unknown global optimum of the fitness function), $A_i=2a\cdot r_{1,i}-a$; $C_i=2r_{2,i}$; with $r_{1,i}$ and $r_{2,i}$ being random variables uniformly distributed within [0, 1] and generated independently for each component i. The parameter a decreases linearly from 2 to 0 as the number of iterations increases, according to

$$\alpha\left(t\right)=2-2t/MaxIter, \qquad (20)$$

where $MaxIter$ denotes the predefined maximum number of iterations.

Since the global optimum is not known, an iterative approximation procedure is applied. Wolves α, β, and δ represent the three best solutions at iteration t. Updated positions for all wolves are calculated at iteration $t+1$. For each wolf in the ω category, auxiliary position vectors are computed using the known positions of α, β, and δ wolves as follows:

$$X_{1,i}^{(k)}\left(t+1\right)=X_i^{(\alpha)}\left(t\right)-$$
$$-A_i^{(\alpha)}\cdot\left|C_i^{(\alpha)}\cdot X_i^{(\alpha)}\left(t\right)-X_i^{(k)}\left(t\right)\right|, \qquad (21)$$

$$X_{2,i}^{(k)}\left(t+1\right)=X_i^{(\beta)}\left(t\right)-$$
$$-A_i^{(\beta)}\cdot\left|C_i^{(\beta)}\cdot X_i^{(\beta)}\left(t\right)-X_i^{(k)}\left(t\right)\right|, \qquad (22)$$

$$X_{3,i}^{(k)}\left(t+1\right)=X_i^{(\delta)}\left(t\right)-$$
$$-A_i^{(\delta)}\cdot\left|C_i^{(\delta)}\cdot X_i^{(\delta)}\left(t\right)-X_i^{(k)}\left(t\right)\right|, \qquad (23)$$

where i denotes the component of the N-dimensional search space and k denotes the index of the ω-category wolf.

The final position of any considered wolf is then determined as

$$\vec{X}(t+1) = \\ = \left(\vec{X}_1(t+1) + \vec{X}_2(t+1) + \vec{X}_3(t+1)\right) \Big/ 3. \quad (24)$$

In the GWO framework, the grey wolf pack performs a parallel search in the solution space to identify a set of optimal UAV swarm positions for each time interval. The algorithm provides an approximate solution that closely approaches the true optimum and operates in a nested loop with the task offloading procedure. The GWO process begins with the initialization of parameters using a chaotic distribution, subject to the imposed constraints and the predefined maximum number of iterations. The initial positions of the wolves, representing candidate UAV placements, are then generated randomly. The fitness of all wolves, including the leaders and followers, is evaluated using function (13). The position of the prey is considered to correspond to the wolf with the highest fitness value. Subsequently, the parameters of GWO are updated, leading to the adjustment of wolf positions. This updating process is repeated until an optimal solution is identified or the maximum number of iterations is reached.

7. Analysis of Simulation Results for the Integrated IoT–MEC–SDN System

The proposed solutions were modeled in the NS-3 environment with integration of the LIMoSim simulator, a lightweight and integrated tool developed based on the widely used INET framework. LIMoSim enables simulations without the need for interaction with external simulators through inter-process communication (IPC) mechanisms and offers several advantages, including integration, reduced configuration complexity, compatibility with INET extensions, and the ability to efficiently handle dynamic scenarios. LIMoSim is implemented on a shared codebase for communication simulation, allowing effective interaction within a single process.

Additionally, the CloudSim environment was employed to evaluate the performance of the proposed solutions.

CloudSim allows modeling of cloud data centers, virtualization, resource management, and workload distribution, providing an assessment of algorithm efficiency in both fog and cloud environments without the need for expensive physical infrastructure. The terrestrial network was represented by an IoT deployment with 2,000 end devices and five independent gateways distributed over a 25 km² area. The aerial network consisted of nine UAVs positioned at a uniform altitude of 45 meters. To evaluate the performance of the proposed solutions for the air–ground distributed edge computing architecture and the optimized model, three system deployment scenarios were considered:

V1 – traditional IoT networks without support for terrestrial or aerial fog computing;

V2 – IoT network with support for distributed terrestrial and aerial fog computing;

V3 – optimized system with an aerial fog layer and IoT network supported by G-MEC and A-MEC servers.

Energy consumption and task processing delay were considered as primary performance metrics.

Energy consumption was evaluated for the three system deployment scenarios. Fig. 2 presents a comparison of average energy consumption (in percentage) for the three considered scenarios at varying numbers of tasks per IoT end device.

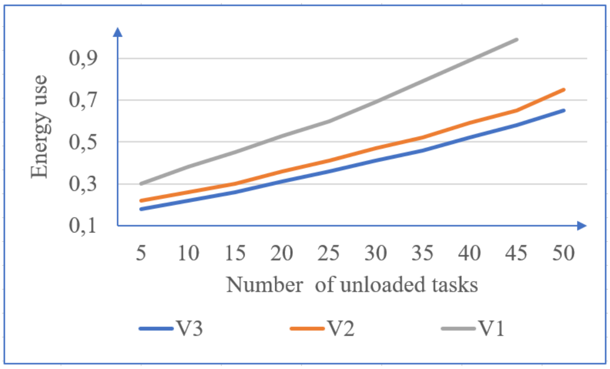

Fig. 2. Average energy consumption at varying numbers of computational tasks (fraction of initial available energy)

These measurements were recorded for 2,000 end devices, with tasks corresponding to the processing of static images. Energy consumption increases as the number of computational tasks per end device grows; however, the proposed air–ground model preserves energy, allowing more tasks to be offloaded compared to traditional IoT networks. The proposed model reduces energy consumption by an average of 20% compared to the traditional IoT model. Moreover, the optimized air–ground IoT network achieves even greater energy savings, with an average efficiency improvement of approximately 5% over the standard air–ground model. The impact of end device density on energy consumption was also analyzed for the three considered deployment scenarios. Fig. 3 shows the percentage of energy consumption for each scenario with varying numbers of end devices, ranging from 1,000 to 3,000 devices.

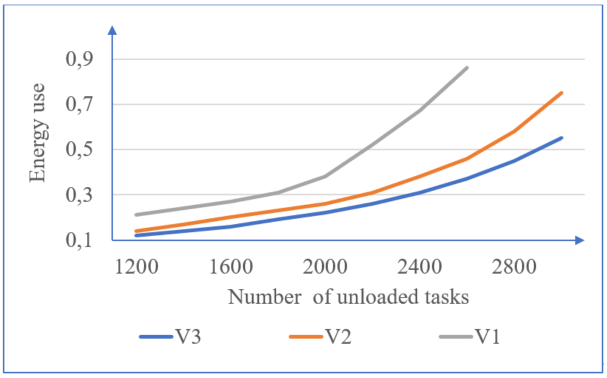

Fig. 3. Average energy consumption for varying numbers of deployed nodes

The tasks considered in this analysis belonged to the second category, with each end device executing ten tasks. The proposed air–ground model reduced energy consumption compared to the traditional network, while the optimized model provided additional reductions. In networks with a high density of deployed end devices, the proposed model significantly reduces energy consumption, achieving more than a twofold improvement compared to densely deployed traditional networks. In all cases, the proposed offloading scheme reduces overall energy consumption.

Latency is another critical network performance metric evaluated during the simulation. The average task processing latency was measured for all three deployment scenarios while varying the number of active end devices. Fig. 4 illustrates the dependence of average processing latency on the number of computational tasks assigned to each end device for the three considered deployment scenarios.

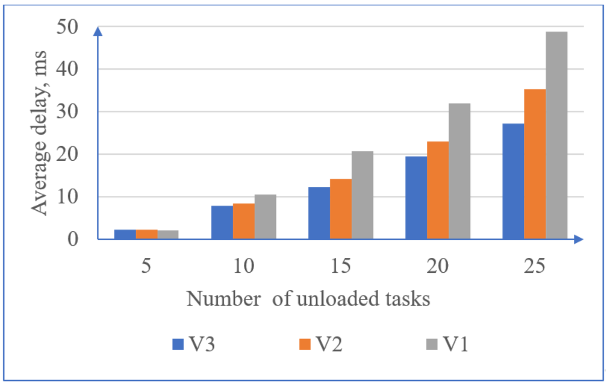

Fig. 4. Average latency incurred during computational task processing

These measurements were obtained for 2,000 end devices executing lightweight computational tasks. The proposed air–ground system (Scenario V2) provides higher latency efficiency compared to traditional systems without fog computing, which rely solely on terrestrial edge computing. This improvement in efficiency increases with the number of offloaded computational tasks. The use of A-MEC servers enables task offloading, thereby reducing the total processing time. Moreover, the optimized system achieves an additional reduction in average latency of up to 3%.

Fig. 5 shows the average task processing latency in each of the three considered system scenarios for five different application complexity categories.

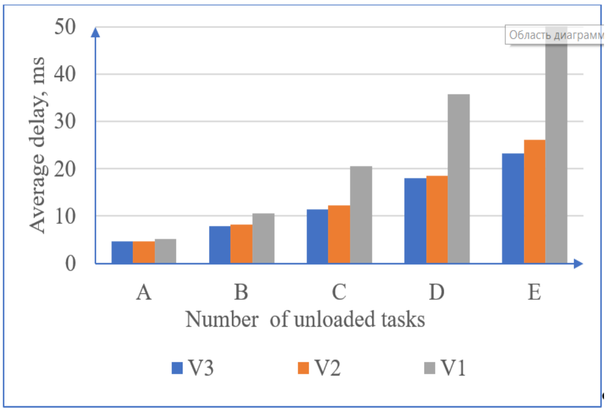

Fig. 5. Average task processing latency for five different application complexity categories

These results were obtained for a network of 2,000 end devices, each executing ten tasks, with gradually increasing task complexity.

Both the proposed air–ground system and the optimized system reduce the average latency required to process computational tasks, particularly for tasks with the highest complexity.

Conclusions

This study investigated the integration of SDN and MEC technologies in support systems for high-density and ultra-high-density Internet of Things (IoT) deployments.

A proposed architecture of the integrated IoT–SDN–MEC system includes both terrestrial and aerial segments.

For this architecture, a mathematical model was developed to estimate the energy consumption of fog-layer devices and the task processing latency.

A traffic offloading scheme within the integrated IoT–MEC–SDN system was described.

The optimization problem of minimizing energy consumption and processing latency for tasks offloaded to the aerial segment was formulated.

To solve this problem, the Grey Wolf Optimizer (GWO) algorithm was applied, providing a rapid approximation of the optimal solution.

Simulation results for the integrated IoT–MEC–SDN system demonstrate that incorporating an aerial fog segment in high-density and ultra-high-density IoT networks reduces both average energy consumption and the average latency incurred during computational task processing.

References

1. Zaman, M., Puryear, N., Abdelwahed, S. and Zohrabi, N. (2024), "A Review of IoT-Based Smart City Development and Management", *Smart Cities*, vol. 7(3), pp. 1462–1501, doi: https://doi.org/10.3390/smartcities7030061
2. Kuchuk, H., Husieva, Y., Novoselov, S., Lysytsia, D. and Krykhovetskyi, H. (2025), "Load Balancing of the layers Iot Fog-Cloud support network", *Advanced Information Systems*, vol. 9, no. 1, pp. 91–98, doi: https://doi.org/10.20998/2522-9052.2025.1.11
3. Wing Lo, Y., Ho Tsoi, M., Chow, C.F. and Mung, S.W.Y. (2025), "An NB-IoT Monitoring System for Digital Mobile Radio With Industrial IoT Performance and Reliability Evaluation", *IEEE Sensors Journal*, vol. 25(3), pp. 5337–5348, doi: https://doi.org/10.1109/JSEN.2024.3512859
4. Kuchuk, N., Kashkevich, S., Radchenko, V., Andrusenko, Y. and Kuchuk, H. (2024), "Applying edge computing in the execution IoT operative transactions", *Advanced Information Systems*, vol. 8, no. 4, pp. 49–59, doi: https://doi.org/10.20998/2522-9052.2024.4.07

5. Alsadie, D. (2024), "Advancements in heuristic task scheduling for IoT applications in fog-cloud computing: challenges and prospects", *PeerJ Computer Science*, 10, e2128, doi: https://doi.org/10.7717/PEERJ-CS.2128
6. Mani Kiran, C.V.N.S., Jagadeesh Babu, B. and Singh, M.K. (2023), "Study of Different Types of Smart Sensors for IoT Application Sensors", *Smart Innovation, Systems and Technologies*, vol. 290, pp. 101–107, doi: https://doi.org/10.1007/978-981-19-0108-9_11
7. Lee, B.M. (2025), "Efficient Resource Management for Massive MIMO in High-Density Massive IoT Networks", *IEEE Transactions on Mobile Computing*, vol. 24(3), pp. 1963–1980, doi: https://doi.org/10.1109/TMC.2024.3486712
8. Zhou, B. and Saad, W. (2024), "Age of Information in Ultra-Dense IoT Systems: Performance and Mean-Field Game Analysis", *IEEE Transactions on Mobile Computing*, vol. 23(5), pp. 4533–4547, doi: https://doi.org/10.1109/TMC.2023.3292515
9. Jang, H.-C. and Li, T.-C. (2024), "Enhancing Edge Computing in High-Density IoT for Improved Service Quality and Privacy Protection", Iet Conference Proceedings, vol. 2024(22), pp. 142–143, doi: https://doi.org/10.1049/icp.2024.4321
10. Maftei, A.A., Petrariu, A.I., Popa, V. and Lavric, A. (2025), "A Blockchain Framework for Scalable, High-Density IoT Networks of the Future", *Sensors*, vol. 25(9), 2886, doi: https://doi.org/10.3390/s25092886
11. Lee, B.M. (2025), "Efficient Resource Management for Massive MIMO in High-Density Massive IoT Networks", IEEE Transactions on Mobile Computing, vol. 24(3), pp. 1963–1980, doi: https://doi.org/10.1109/TMC.2024.3486712
12. Kuchuk, H., Kalinin, Y., Dotsenko, N., Chumachenko, I. and Pakhomov, Y. (2024), "Decomposition of integrated high-density IoT data flow", *Advanced Information Systems*, vol. 8, no. 3, pp. 77–84, doi: https://doi.org/10.20998/2522-9052.2024.3.09
13. Zhang, Y., Jing, R., Zou, Y. and Cao, Z. (2025), "Optimizing power allocation in contemporary IoT systems: A deep reinforcement learning approach. Sustainable Computing: Informatics and Systems", vol. 46, number 10114, doi: https://doi.org/10.1016/j.suscom.2025.101114
14. Dankolo, N.M., Radzi, N.H.M., Mustaffa, N.H., Arshad, N. I., Nasser, M., Gabi, D. and Yusuf, M.N. (2025), "Optimizing resource allocation for IoT applications in the edge cloud continuum using hybrid metaheuristic algorithms", *Scientific Reports*, vol. 15(1), 14409, doi: https://doi.org/10.1038/s41598-025-97648-2
15. Kuchuk, H., Mozhaiev, O., Tiulieniev, S., Mozhaiev, M., Kuchuk, N., Tymoshchyk, L., Lubentsov, A., Onishchenko, Y., Gnusov, Y. and Tsuranov, M. (2025), "Devising a method for increasing data transmission speed in monitoring systems based on the mobile high-density Internet of Things", *Eastern-European Journal of Enterprise Technologies*, *3*(4 (135)), pp. 52–61, doi: https://doi.org/10.15587/1729-4061.2025.330644
16. Hudda, S. and Haribabu, K. (2025), ‘'A review on WSN based resource constrained smart IoT systems", *Discover Internet of Things*, vol. 5(1), 56, doi: https://doi.org/10.1007/s43926-025-00152-2
17. Al-Hammadi, I., Li, M. and Islam, S.M.N. (2023), "Independent tasks scheduling of collaborative computation offloading for SDN-powered MEC on 6G networks", *Soft Computing*, vol. 27(14), pp. 9593–9617, doi: https://doi.org/10.1007/s00500-023-08091-2
18. Kalinin, Y., Kozhushko, A., Rebrov, O. and Zakovorotniy, A. (2022), "Characteristics of Rational Classifications in Game-Theoretic Algorithms of Pattern Recognition for Unmanned Vehicles", *2022 IEEE 3rd Khpi Week on Advanced Technology Khpi Week*, 2022 Conference Proceedings, doi: https://doi.org/10.1109/KhPIWeek57572.2022.9916454
19. Kovalenko, A., Kuchuk, H., Radchenko, V. and Poroshenko, A. (2020), "Predicting of Data Center Cluster Traffic", *2020 IEEE International Conference on Problems of Infocommunications Science and Technology*, PIC S and T 2020 – Proceedings, pp. 437–441, 9468006, doi: https://doi.org/10.1109/PICST51311.2020.9468006
20. Kuchuk, H., Mozhaiev, O., Tiulieniev, S., Mozhaiev, M., Kuchuk, N., Tymoshchyk, L., Onishchenko, Yu., Tulupov, V., Bykova, T. and Roh, V. (2025), "Devising a method for forming a stable mobile cluster of the internet of things fog layer", *Eastern-European Journal of Enterprise Technologies*, 2025, vol. 1, no. 4(133), pp. 6–14, doi: https://doi.org/10.15587/1729-4061.2025.322263
21. Kuchuk, N., Zakovorotnyi, O., Pyrozhenko, S., Radchenko, V. and Kashkevich, S. (2025), "A method for redistributing virtual machines of heterogeneous data centres", *Advanced Information Systems*, vol. 9, no. 1, pp. 80–85, doi: https://doi.org/10.20998/2522-9052.2025.1.09
22. Kuchuk, N., Mozhaiev, M., Kalinin, Y., Mozhaev, O. and Kuchuk, H. (2022), "Calculation of Signal Information Delay in Intelligent Communication Networks", *2022 IEEE 3rd KhPI Week on Advanced Technology*, KhPI Week 2022 - Conference Proceedings, doi: https://doi.org/10.1109/KhPIWeek57572.2022.9916323
23. Xu, C. (2025), "Resource optimization algorithm for 5G core network integrating NFV and SDN technologies", *International Journal of Intelligent Networks*, vol. 6, pp. 36–46, doi: https://doi.org/10.1016/j.ijin.2025.04.001
24. Devagiri, S. and Vijayalakshmi, M. (2025), "Surveying Advancements of 5G Technology and Network Evolution with MEC Integration", *Lecture Notes in Networks and Systems*, vol. 1159, pp. 357–366, doi: https://doi.org/10.1007/978-981-97-8526-1_28
25. Hunko, M., Tkachov, V., Kuchuk, H., Kovalenko, A. (2023), Advantages of Fog Computing: A Comparative Analysis with Cloud Computing for Enhanced Edge Computing Capabilities, *2023 IEEE 4th KhPI Week on Advanced Technology, KhPI Week 2023 – Conf. Proceedings*, 02-06 October 2023, Code 194480, doi: https://doi.org/10.1109/KhPIWeek61412.2023.10312948
26. Kuchuk, H., Mozhaiev, O., Tiulieniev, S., Mozhaiev, M., Kuchuk, N., Tymoshchyk, L., Lubentsov, A., Gnusov, Y., Klivets, S. and Kuleshov, A. (2025), "Devising a method for stabilizing control over a load on a cluster gateway in the internet of things edge layer", *Eastern-European Journal of Enterprise Technologies*, vol. *2*(9 (134)), pp. 24–32, doi: https://doi.org/10.15587/1729-4061.2025.326040
27. Zakovorotniy, A., and Kharchenko, A. (2021), "Optimal speed controller design with interval type-2 fuzzy sets", *2021 IEEE 2nd Khpi Week on Advanced Technology Khpi Week 2021 Conference Proceedings* , pp. 363–366, doi: https://doi.org/10.1109/KhPIWeek53812.2021.9570045

28. Kuchuk, N., Mozhaiev, O., Semenov, S., Haichenko, A., Kuchuk, H., Tiulieniev, S., Mozhaiev, M., Davydov, V., Brusakova, O. and Gnusov, Y. (2023), "Devising a method for balancing the load on a territorially distributed foggy environment", *Eastern-European Journal of Enterprise Technologies*, vol. 1(4 (121), pp. 48–55, doi: https://doi.org/10.15587/1729-4061.2023.274177
29. Wang, Y., Wei, Z., Huang, Z., Yang, J. and Zhao, J. (2025), "Dependent task offloading for air-ground integrated MEC networks: a multi-agent collaboration approach", *Cluster Computing*, vol. 28(2), 129, doi: https://doi.org/10.1007/s10586-024-04732-9
30. Petrovska, I., Kuchuk, H., Kuchuk, N., Mozhaiev, O., Pochebut, M. and Onishchenko, Yu. (2023), "Sequential Series-Based Prediction Model in Adaptive Cloud Resource Allocation for Data Processing and Security", *2023 13th International Conference on Dependable Systems, Services and Technologies*, DESSERT 2023, 13–15 October, Athens, Greece, code 197136, doi: https://doi.org/10.1109/DESSERT61349.2023.10416496
31. Izadi, M., Mohammad-Khani, G.-R. and Farahani, G. (2025), "Improving the performance of the MIMO-OFDM-NOMA System using a V-BLAST ZF approach based on deep CNN in IoT", *Eurasip Journal on Wireless Communications and Networking*, vol. 2025(1), 51, doi: https://doi.org/10.1186/s13638-025-02481-w
32. Kuchuk, G., Nechausov, S. and Kharchenko, V. (2015), "Two-stage optimization of resource allocation for hybrid cloud data store", *International Conference on Information and Digital Technologies*, *Zilina*, pp. 266–271, DOI: http://dx.doi.org/10.1109/DT.2015.7222982
33. Roy, S., Mazumdar, N. and Pamula, R. (2025), "A multi-depot provisioned UAV swarm trajectory optimization scheme for collaborative data acquisition in a large-scale IoT environment", *Ad Hoc Networks*, vol. 178, 103974, doi: https://doi.org/10.1016/j.adhoc.2025.103974
34. Ragulin, V., Owaid, S.R., Kuchuk, H., Gaman, O. and Hurskyi, T. (2024), "Development of a method for increasing the efficiency of processing heterogeneous data using a metaheuristic algorithm", *Eastern-European Journal of Enterprise Technologies*, vol. 4(3(130)), pp. 21–28, doi: https://doi.org/10.15587/1729-4061.2024.309126
35. Tarkhan, A.B., Kuchuk, H., Stanovska, I., Zvershkhovskyi, I. and Fysiuk, A. (2024), "Development of an evaluation method using a combined cat swarm optimization algorithm", *Eastern-European Journal of Enterprise Technologies*, vol. 3(4(129)), pp. 55–63, doi: https://doi.org/10.15587/1729-4061.2024.305363
36. Sharma, S. and Kapoor, A. (2021), "An efficient routing algorithm for IoT using GWO approach", *International Journal of Applied Metaheuristic Computing*, vol. 12(2), pp. 67–84, doi: https://doi.org/10.4018/IJAMC.2021040105

Received (Надійшла) 11.06.2025
Accepted for publication (Прийнята до друку) 10.09.2025

ВІДОМОСТІ ПРО АВТОРІВ/ ABOUT THE AUTHORS

Шефер Олександр Віталійович – доктор технічних наук, професор, завідувач кафедри автоматики, електроніки та телекомунікацій, Національний університет «Полтавська політехніка імені Юрія Кондратюка», Полтава, Україна;
Oleksandr Shefer – Doctor of Technical Sciences, Professor, Head of the Department of Automation, Electronics and Telecommunications, National University "Yuri Kondratyuk Poltava Polytechnic", Poltava, Ukraine;
e-mail: itm.ovshefer@nupp.edu.ua; ORCID Author ID: https://orcid.org/0000-0002-3415-349X;
Scopus ID: https://www.scopus.com/authid/detail.uri?authorId=57210203269.

Мигаль Станіслав Вікторович – аспірант кафедри автоматики, електроніки та телекомунікацій, Національний університет «Полтавська політехніка імені Юрія Кондратюка», Полтава, Україна;
Stanislav Myhal – PhD student of the Department of Automation, Electronics and Telecommunications, National University "Yuri Kondratyuk Poltava Polytechnic", Poltava, Ukraine;
e-mail: stas.migal1998@gmail.com; ORCID Author ID: https://orcid.org/0009-0007-4675-7358.

Метод побудови мережі підтримки високощільного Інтернету речей з інтеграцією технологій SDN та MEC

О. В. Шефер, С. В. Мигаль

Анотація. Актуальність. При впровадженні IoT високої та надвисокої щільності виникає низка технічних та організаційних проблем. Одним з напрямів подолання цих проблем є інтеграція технологій опрацювання мереж та даних, зокрема SDN TA MEC. Отже, **метою даної статті** є дослідження процесу інтеграції технологій SDN TA MEC в системи підтримки високощільного та надвисокощільного Інтернету речей. **Отримані результати.** Запропонована архітектура інтегрованої системи IoT-SDN-MEC, що складається з наземного та повітряного сегментів. Для даної архітектури розроблена математична модель, котра дозволяє визначати енергетичні витрати пристроїв туманного шару та часові затримки виконання завдань. Описана схема вивантаження трафіка в інтегрованій системі IoT-MEC-SDN. Сформульовано завдання оптимізація енергоспоживання та затримок обробки завдань, вивантажених у повітряний сегмент. Для його вирішення застосований алгоритм зграї сірих вовків, який допомагає швидко знайти наближений розв'язок. **Висновок.** Проведений аналіз результатів моделювання функціонування інтегрованої системи IoT-MEC-SDN показав, що при використанні у туманному шарі повітряного сегменту для високощільного та надвисокощільного Інтернету речей зменшуються середні енерговитрати та середня затримка, що виникає при обробці обчислювальних задач. **Подальші дослідження** у даному напрямі пов'язані з формуванням оптимальної структури повітряного сегменту.

Ключові слова: Інтернет речей; технологія SDN; технологія MEC; високощільний; надвисокощільний; інтегрована система; наземний сегмент, повітряний сегмент.