

Oleksandr Shefer, Stanislav Myhal

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

## DEVELOPMENT OF AN ARCHITECTURAL DESIGN METHOD FOR MOBILE SDN IN ULTRA-DENSE SENSOR NETWORKS

**Abstract. Background.** The method of deploying SDN controllers on mobile nodes within ultra-dense networks is a vital component in enhancing the management efficiency of modern telecommunication systems. Given the rapid proliferation of connected devices and escalating traffic volumes, traditional centralized network management approaches often lack the necessary flexibility and scalability. **Subject of Research.** This study focuses on methods for constructing mobile SDN architectures tailored for ultra-dense sensor networks (UDSNs). **Objectives.** The aim of this research is to develop a method for designing a software-defined ultra-dense sensor network architecture, where specific SDN controllers are integrated into mobile nodes at the edge layer. **Results.** The study proposes an MEC (Multi-access Edge Computing) platform architecture specifically designed for deploying SDN controllers on mobile nodes within 5G/6G ultra-dense networks. A three-tier model for a mobile multi-controller SDN in UDSNs has been established. Furthermore, a specialized method for the placement and distribution of SDN controllers on mobile nodes within these ultra-dense environments was developed. **Conclusion.** Experimental results indicate that the implementation of the proposed method reduces transaction latency by 60% compared to traditional SDN-based edge computing models. Additionally, energy consumption is reduced by up to 30%.

**Keywords:** telecommunication network, edge layer, mobile device, controller, ultra-dense sensor network (UDSN), OpenFlow switch, 5G standard, SDN.

### Introduction

The modern advancement of wireless technologies and the Internet of Things (IoT) has led to a rapid increase in the number of sensor devices operating within unified network infrastructures [1].

Ultra-dense sensor networks (UDSNs) are becoming a pivotal element in domains such as smart cities, the Industrial Internet of Things (IIoT), environmental monitoring, and autonomous transport systems [2, 3]. Simultaneously, the increasing network density complicates management, intensifies resource load, and presents new challenges regarding scalability and energy efficiency [4].

Traditional networking approaches fail to provide the necessary flexibility and adaptability required in such dynamic environments.

One of the most promising directions for addressing these issues is the implementation of the Software-Defined Networking (SDN) concept, which entails the decoupling of the control plane from the data plane [5, 6]. This approach enables centralized network management and enhances overall controllability. However, in the context of ultra-dense sensor networks, a strictly centralized SDN architecture may lead to controller overhead and increased latency.

Consequently, decentralized and hierarchical SDN architectures, as well as approaches involving mobile nodes, are gaining significant attention. Integrating mobility into the network's edge layer opens new possibilities for optimizing controller placement and improving data processing efficiency [7, 8]. Mobile SDN controllers are capable of adapting to changes in network topology and reducing latency by maintaining proximity to data sources.

Despite a substantial body of research in this field, the issues of constructing an effective mobile SDN architecture for ultra-dense sensor networks remain insufficiently explored. In particular, methods for optimal controller placement, load balancing, and

ensuring network resilience require further analysis. Furthermore, it is crucial to account for the constrained resources of sensor nodes.

Thus, the development of a mobile SDN architecture for ultra-dense sensor networks represents a significant step toward improving the reliability and operational efficiency of modern software-defined networks.

### 1. Literature Review

Contemporary research focuses significantly on the evolution of the Software-Defined Networking (SDN) concept as a foundation for building flexible and scalable network infrastructures. In [9, 10], a fundamental analysis of SDN is presented, defining key principles such as the decoupling of the control and data planes. This separation facilitates centralized network management and simplifies configuration; however, the authors also identify scalability and reliability issues inherent in using centralized controllers within large-scale networks.

Subsequent studies have pivoted toward the application of SDN in wireless sensor networks (WSNs). Research in [11] provides an overview of employing SDN to enhance management efficiency in sensor networks, particularly regarding routing optimization and reduced energy consumption. The authors emphasize that traditional management approaches fail to meet the rigorous requirements of ultra-dense environments.

The architectural intricacies of integrating SDN into wireless sensor networks are explored in [12]. While the proposed solutions improve network flexibility and ensure effective interaction between nodes and controllers, the challenge of optimal resource allocation remains unresolved.

The study in [13] is dedicated to developing an SDN architecture for 6LoWPAN networks, enabling the integration of sensor devices into IP-oriented environments. This represents a significant step toward building scalable IoT systems; nevertheless, the problem

of adapting to topological changes necessitates further investigation.

In [14], the problem of SDN controller placement in 5G-IoT networks is addressed. The proposed approach aims to minimize data transmission latency and ensure load balancing, both of which are critical for ultra-dense sensor networks.

The use of SDN for constructing heterogeneous networks, including nano-networks, is proposed in [15], demonstrating the potential for scaling and integrating diverse devices. This confirms the promise of SDN as a universal technology for managing complex network systems.

Research in [16] examines the integration of intelligent mechanisms into next-generation networks, specifically 5G and IoT. The author underscores the importance of combining SDN with artificial intelligence (AI) technologies to enhance network adaptability and autonomy.

The analysis of current research [9–16] indicates active progress in developing SDN-based approaches for sensor and IoT networks. However, existing solutions are primarily oriented toward static or semi-dynamic environments and do not sufficiently account for node mobility. This gap necessitates the development of new methods for constructing mobile SDN architectures for ultra-dense sensor networks, which constitutes the subject of this study.

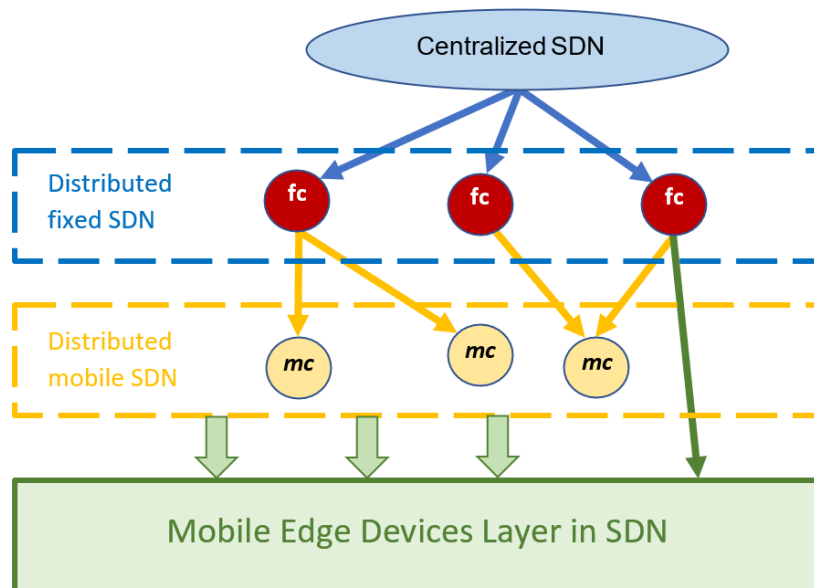
**Research Objective:** To develop and evaluate a method for constructing a software-defined ultra-dense sensor network architecture, wherein specific SDN controllers are deployed on mobile nodes at the edge layer.

## 2. Mobile SDN Structure

The evolution of SDN technology has progressed from a centralized scheme with a single SDN controller toward a distributed framework utilizing multiple controllers. The architecture of such networks has been established based on distributed network management through edge computing nodes [17]. In these networks, SDN controller functions are distributed across edge computing nodes to implement a centralized management scheme [18]. Each virtual controller interfaces with a centralized management framework that supports one or more SDN controllers, depending on the network scale.

Modifications to SDN technology are aimed at providing SDN controller mobility to support high-density and ultra-dense network scenarios. A variant of the network model featuring a mobile SDN controller is depicted in Fig. 1. In this context, the SDN network can be categorized into three primary levels:

- Centralized SDN controllers;
- Distributed stationary SDN controllers;
- Distributed mobile SDN controllers.



**Fig. 1.** Network model with a mobile SDN controller

**Centralized SDN Controllers.** Several centralized SDN controllers are utilized to construct the SDN network, providing overarching management. This management method defines the primary level of network control and maintains the necessary interfaces with network operators.

Previously proposed algorithms are employed to address the challenges of controller placement and distribution within the network. In this case, the network architecture consists of the following six core components.

*The Control Plane* is a set of network applications that govern the logic of the SDN network. Software tools are utilized to ensure flexibility and ease of deployment for new applications and services, such as routing, load balancing, policy enforcement, or user applications. They also facilitate the orchestration and automation of network operations through existing Application Programming Interfaces (APIs).

*Controllers* represent the most intelligent and critical layer of the SDN architecture, containing one or more controllers that transmit various types of rules and

policies to the infrastructure layer via the Southbound (SB) interface.

The Data Plane (*Infrastructure Layer*) consists of a set of data forwarding devices within the network (routers, switches, load balancers, etc.). It utilizes Southbound APIs (SB APIs) to interact with the control plane, receiving data forwarding rules and policies to be applied to the respective devices.

Northbound (NB) Interfaces provide communication between the control and management layers, typically comprising a set of open-source APIs.

East-West Interfaces enable communication between multiple controllers. They utilize messaging systems or distributed routing protocols, such as BGP (Border Gateway Protocol) and OSPF (Open Shortest Path First). These interfaces are also used for communication between centralized controllers and other network devices.

Southbound Interfaces facilitate interaction between the control plane and the data plane; these are protocols that allow the controller to push policies to the forwarding plane. The OpenFlow protocol is the most widely recognized and prevalent protocol for SDN-enabled networks. However, OpenFlow is only one of

many SDN protocols; others include OpFlex, which redistributes certain network management tasks to the infrastructure layer to improve scalability, and ForCES, which offers a flexible method for enhancing the management of traditional networks without a logically centralized controller. Additionally, the ROFL (Revised OpenFlow Library) provides an API for software developers to more efficiently create new applications.

**Stationary Distributed SDN Controllers.** The layer of stationary distributed SDN controllers includes an integrated SDN controller at the edge of the Radio Access Network (RAN). The RAN consists of distributed heterogeneous cellular cells with eNB base stations.

Each eNB is connected to a Multi-Access Edge Computing (MEC) server. In this configuration, the MEC represents a multi-tier structure, as the utilized MEC servers are heterogeneous in terms of computing capabilities and feature a hierarchical structure of three primary cloud tiers interconnected by high-speed fiber-optic links [19].

A new architecture for the existing MEC-based RAN in 5G/6G networks is developed, as shown in Fig. 2, employing a multi-level interaction system of edge computing systems, as illustrated in Fig. 1

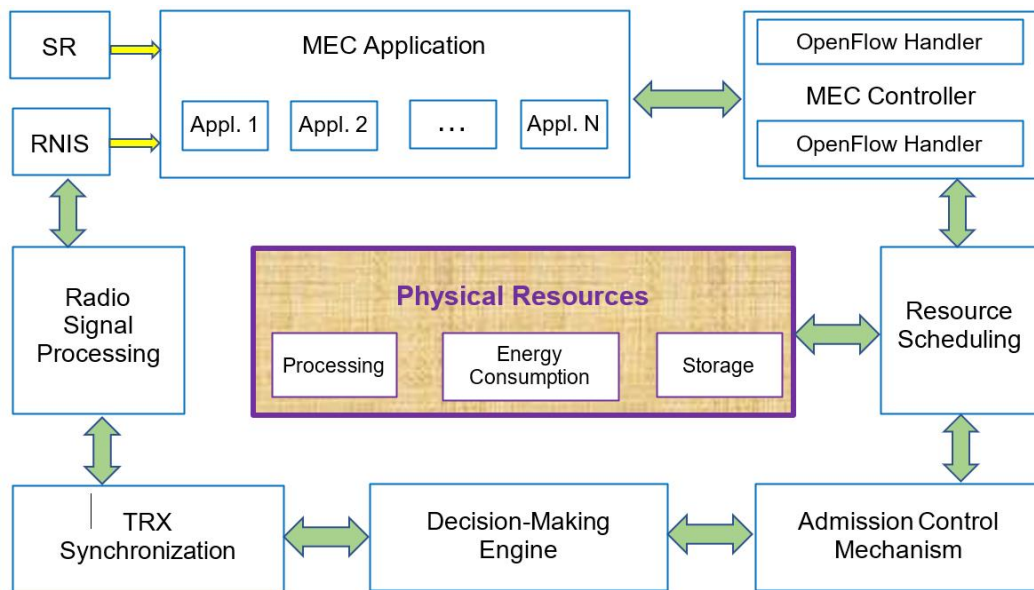


Fig. 2. MEC platform architecture for the SDN controller placement method on mobile nodes in 5G/6G ultra-dense networks

The first core component of the MEC platform is the hardware or physical resources, comprising the CPU, storage units, and hardware tools used for power delivery and process monitoring. This component varies across different cloud tiers. Hardware resources are either utilized or shared between local and offloaded computing tasks.

To ensure the efficient use of edge computing resources, a resource scheduler is implemented to manage and allocate resources among various computational tasks.

Computational tasks are offloaded to the MEC server, and the system must decide, based on available resources and predefined Quality of Service (QoS)

criteria, whether to process the offloaded task. Consequently, the MEC platform must include a decision-making mechanism that responds to offloading requests by either accepting or rejecting application offloading tasks. The decision-making mechanism is the part of the MEC platform that implements application offloading algorithms. To make an offloading decision, this mechanism requires data on the currently available server resources, which is provided via the admission control system.

Another component of the platform is the synchronizer/TRX (Transceiver), a hardware-software system responsible for connecting the MEC server with other devices, such as eNBs. Thus, the synchronizer/TRX

facilitates the transmission and reception of data to and from the MEC platform.

Each MEC server features a controller that hosts a Virtual Machine (VM) implementing the SDN network management method used in the core network. This MEC controller facilitates integration and interaction with the core network. Furthermore, the MEC controller manages and monitors server operations to ensure high availability, flexibility, and reduced latency. The MEC controller receives updates from the core SDN controller via the OpenFlow protocol and updates the relevant components of the MEC platform accordingly. The communication between MEC servers and the SDN controller via OpenFlow ensures efficient and accelerated interaction between the SDN controller and the RAN.

Additionally, MEC applications provide network operators with new avenues for deploying MEC-based services.

MEC-based stationary distributed SDN controllers interface with both centralized and mobile SDN controllers.

**Mobile Distributed SDN Controllers.** These controllers are typically lightweight SDN controllers deployed on moving objects. There are numerous use cases for mobile SDN controllers; for instance, they can be installed on vehicles to support interaction within ultra-dense networks and ensure ultra-high system availability. A mobile SDN controller possesses one or more interfaces with stationary distributed SDN controllers. This integration fulfills the connectivity and availability requirements of 6G networks. A lightweight MEC platform architecture (Fig. 2) is utilized for the implementation of mobile controllers.

Furthermore, mobile controllers provide localized decision-making, which significantly reduces latency in processing network events.

This is particularly critical for scenarios with stringent response time requirements, such as autonomous vehicles or the Tactile Internet. The use of mobile SDN controllers also helps alleviate the load on central control nodes by distributing network management functions. Moreover, these controllers can dynamically adapt to changes in network topology caused by the movement of users or infrastructure nodes.

A key advantage is the ability to integrate with MEC resource orchestration mechanisms, ensuring the efficient utilization of computational and network resources at the periphery.

Mobile controllers can also perform data pre-processing and traffic filtering, reducing the volume of information transmitted to central data centers. In terms of energy efficiency, they allow for optimized power consumption through local load management.

However, the deployment of mobile distributed SDN controllers introduces several challenges, such as maintaining network state consistency among controllers and ensuring reliable synchronization. Security is equally paramount, as mobile nodes may be more susceptible to attacks. Consequently, research in this field for ultra-dense networks must focus on developing effective coordination mechanisms, security protocols, and

adaptive management for mobile SDN controllers in next-generation networks.

### 3. Model of a Mobile Multi-Controller SDN for Ultra-Dense Sensor Networks

To analyze the proposed system, we construct a mobile multi-controller SDN model using graph theory. In SDN environments, in-band communication is typically employed as the primary signaling method.

**Formalization of Network Connections.** The controller network topology is represented as an undirected graph  $G = (V, E)$ , where  $V$  and  $E$  denote the sets of nodes and the links between them, respectively.

The set of SDN controllers deployed in the core network is denoted as  $N_C$  and defined as follows:

$$N_C = \{C_1, C_2, \dots, C_n, \dots, C_N\} \quad \forall n \in \overline{1, N}, \quad (1)$$

where  $N$  is the total number of core SDN controllers.

The set of controllers deployed at the network edge is denoted as  $E_C$  and defined as:

$$E_C = \{C_1, C_2, \dots, C_k, \dots, C_K\} \quad \forall k \in \overline{1, K}, \quad (2)$$

where  $K$  is the total number of SDN controllers deployed at the RAN edge.

Each group of edge controllers interfaces with a core network controller. The set of edge controllers associated with a core controller  $C_n$ , is defined as:

$$E_C^{(n)} = \{C_1^{(n)}, C_2^{(n)}, \dots, C_{kn}^{(n)}, \dots, C_{Kn}^{(n)}\},$$

$$Kn < K, E_C^{(n)} \subset E_C, \forall kn \in \overline{1, Kn}, \quad (3)$$

$$\sum_{n=1}^N Kn = K,$$

where  $Kn$  is the total number of edge controllers interfacing with the core controller  $C_n$ .

The set of deployed mobile controllers is defined as:

$$M_C = \{C_1^{(m)}, C_2^{(m)}, \dots, C_\ell^{(m)}, \dots, C_L^{(m)}\} \quad \forall \ell \in \overline{1, L}, \quad (4)$$

where  $L$  is the total number of deployed mobile SDN controllers.

Each group of mobile controllers interfaces with an edge controller.

The set of mobile controllers associated with an edge controller  $C_k$ , is defined as:

$$M_C^{(k)} = \{C_1^{(m,k)}, C_2^{(m,k)}, \dots, C_{lk}^{(m,k)}, \dots, C_{Lk}^{(m,k)}\},$$

$$Lk < L, M_C^{(k)} \subset M_C, \forall lk \in \overline{1, Lk}, \quad (5)$$

$$\sum_{k=1}^K Lk = L,$$

where  $Lk$  is the total number of mobile controllers interfacing with the edge controller  $C_k$ .

In the data plane, each switch distributed among the controllers is connected to an SDN controller. These connections are determined by a specialized controller placement algorithm. The set of deployed switches is defined as:

$$S = \{S_1, S_2, \dots, S_r, \dots, S_R\} \quad \forall r \in \overline{1, R}, \quad (6)$$

To evaluate controller performance, the response time is used as a metric, which is significantly influenced by queuing delays. Controllers are modeled using an M/M/s multi-server queuing model, where  $s$  is the number of serving devices. The average response time  $T_i$  of controller  $C_i$  is calculated as the sum of the queuing wait time and the processing time. This can be derived using the Erlang-C formula as a function of the request arrival rate  $\lambda_i$  and the service rate  $\mu$ :

$$T_i(\lambda) = C(s, \lambda_i/\mu) / (s \cdot \mu_i - \lambda_i) + \mu^{-1}, \quad (7)$$

where  $C(s, \lambda_i/\mu)$  represents the probability that all servers in the system are busy and any incoming packet will be queued. This probability is calculated as:

$$C\left(s, \frac{\lambda_i}{\mu}\right) = \frac{\left(\frac{(s\rho)^s}{s!}\right) \cdot \left(\frac{1}{1-\rho}\right)}{\sum_{i=0}^{s-1} \frac{(s\rho)^i}{i!} + \left(\frac{(s\rho)^s}{s!}\right) \cdot \left(\frac{1}{1-\rho}\right)} = \frac{1}{1 + \left(\frac{1}{1-\rho}\right) \cdot \left(\frac{s!}{(s\rho)^s}\right) \cdot \sum_{i=0}^{s-1} \frac{(s\rho)^i}{i!}}, \quad (8)$$

where  $\rho = \frac{\lambda_i}{s \cdot \mu}$  is the server utilization factor, serving as an indicator of system stability.

The arrival rate  $\lambda_i$  for the controller is calculated as the sum of the average request rates from the switches connected to that controller.

**Interaction Costs Between Switches and Controllers.** Regarding the interaction between switches and controllers, when a flow table (e.g., a new flow entry) must be established upon request, the switch sends packets to the controller. The controller then calculates the data path and installs the corresponding flow label on the switch. Subsequently, the switch forwards packets according to the flow table. In this process, the total packet latency for controller  $C_n$ , consists of the time required to deliver the packet information to the controller and the controller's subsequent response to the switches. The interaction cost between switch  $S_r$  and controller  $C_n$  in an OpenFlow network is defined as:

$$C_{C-S} = 2 \cdot \sum_{n=1}^N \sum_{r=1}^R p_{\xi-S_r} \left( \frac{\lambda_{S_r}^{(t)}}{p_{\xi}} \cdot d_{C_n-S_r} \cdot B_{C_n-S_r} \right), \quad (9)$$

where  $C_{C-S}$  – the total communication cost between switches and controllers;

$p_{\xi-S_r}$  – the average polling rate of switch  $S_r$ ;

$\lambda_{S_r}^{(t)}$  – the request intensity from switch  $S_r$  over time interval  $t$ ;

$p_{\xi}$  – the specific polling rate of the switch;

$d_{C_n-S_r}$  – the distance between controller  $C_n$  and switch  $S_r$ ;

$B_{C_n-S_r}$  – a boolean variable representing the decision of switch  $S_r$  to connect to controller  $C_n$ , obtained via a Salp Swarm Evolutionary Algorithm.

The interaction cost is determined by the total latency..

To determine the interaction cost between controllers, it is noted that in a multi-controller environment, synchronization of information transfer must be implemented. This ensures that each controller maintains a global view of the network state in near-real-time. The state synchronization cost primarily pertains to the interaction overhead determined by the exchange of state information between controllers across the three aforementioned deployment tiers. This latency is defined as the interaction and synchronization delay between entities within the control plane. The synchronization cost is calculated as follows:

$$C_{C-C} = \alpha 1 \cdot \sum_{n1=1}^N \sum_{n2=1}^N d_{n1,n2}^{(1)} + \alpha 2 \cdot \sum_{n=1}^N \sum_{k=1}^K d_{n,k}^{(2)} + \alpha 3 \cdot \sum_{k=1}^K \sum_{\ell=1}^L d_{k,\ell}^{(3)}, \quad (10)$$

where  $\alpha 1$  – the average rate of state information transfer between controllers in the core network;

$\alpha 2$  – the average rate of state information transfer for edge controllers;

$\alpha 3$  – the average rate of state information transfer for mobile controllers;

$d_{n1,n2}^{(1)}$  – the distance between controllers  $C_{n1}$  and  $C_{n2}$ , located in the core network;

$d_{n,k}^{(2)}$  – the distance between core controller  $C_n$  and edge controller  $C_k$ ;

$d_{k,\ell}^{(3)}$  – the distance between edge controller  $C_k$  and mobile controller  $C_\ell$ .

#### 4 Method for SDN Controller Placement on Mobile Nodes in Ultra-Dense Networks

The proposed method focuses on the adaptive placement of SDN controllers, accounting for the current network state. It is based on the analysis of key Quality of Service (QoS) metrics, specifically latency, throughput, load levels, and link reliability. Furthermore, the method incorporates node mobility, which enables a reduction in the number of network reconfigurations and enhances operational stability. A critical aspect involves minimizing controller migration costs between nodes and reducing the overhead associated with the exchange of control information.

Particular attention is paid to the cost-related aspects of network construction, specifically optimizing the deployment and operational expenditures of the proposed three-tier hierarchical SDN controller structure. This structure allows for effective load balancing among

controllers and minimizes latency. Cost optimization is achieved through the rational selection of the number of controllers at each tier, their spatial distribution, and resource provisioning. In doing so, expenses for computational resources, energy consumption, communication channels, and infrastructure maintenance are considered.

Thus, the proposed approach ensures not only improved efficiency and resilience of ultra-dense networks but also the economic feasibility of their implementation.

The total interaction cost in the network represents the sum of the interaction costs between switches and controllers, as well as the interaction costs between controllers of different tiers at a given point in time. For a specific network topology, the interaction cost between switches and controllers decreases as the number of controllers increases; however, this simultaneously increases the interaction cost among the controllers themselves.

The ultimate goal of the optimization task is to partition the network into management clusters and assign switches to each cluster in a manner that minimizes the overall latency. Consequently, to achieve this objective, the following optimization problem is formulated:

1. Objective Function: On the set of all possible distributions  $\wp = \{\odot\}$  find the minimum of the cost function based on the proposed interaction cost expressions (9) and (10):

$$C = \beta_1 \cdot C_{C-C} + \beta_2 \cdot C_{C-S} \xrightarrow{\gamma \in \Gamma} \min, \quad (11)$$

where  $\beta_2$  – the weighting coefficient for the switch-to-controller interaction cost;

$\beta_1$  – the weighting coefficient for the inter-controller interaction cost.

A normalization rule is applied to these coefficients, such that  $\beta_1 + \beta_2 = 1$ , or  $\beta_2 = 1 - \beta_1$ .

2. Optimization Problem Constraints:

$$B_{C_n-S_r} \in \{0; 1\}; \quad (12)$$

$$T_{end}^{(n-k-\ell)} - T_{begin}^{(n-k-\ell)} \leq \tau_\varphi \quad (13)$$

$$\forall (C_\ell \rightarrow C_k \rightarrow C_n);$$

$$U_{min}^{(1)} \leq U_n^{(1)} \leq U_{max}^{(1)} \quad \forall C_n \in N_C; \quad (14)$$

$$U_{min}^{(2)} \leq U_k^{(2)} \leq U_{max}^{(2)} \quad \forall C_k \in E_C; \quad (15)$$

$$U_{min}^{(3)} \leq U_\ell^{(3)} \leq U_{max}^{(3)} \quad \forall C_\ell^{(m)} \in M_C, \quad (16)$$

where  $T_{begin}^{(n-k-\ell)}$  and  $T_{end}^{(n-k-\ell)}$  are the start and end times, respectively, for the transmission of transaction  $\varphi$  between SDN controllers along the route from  $C_\ell^{(m)}$  to  $C_n$  via  $C_k$ ;

$\tau_\varphi$  – is the maximum allowable transmission time for transaction  $\varphi$ , that ensures the required Quality of Service (QoS) requirements;

$U_n^{(1)}$ ,  $U_k^{(2)}$  and  $U_\ell^{(3)}$  are the utilization indices of SDN controllers corresponding to the core, edge, and mobile tiers, respectively;

$U_{min}^{(1)}$ ,  $U_{min}^{(2)}$  та  $U_{min}^{(3)}$  – are the minimum allowable utilization indices of SDN controllers at the respective tiers;

$U_{max}^{(1)}$ ,  $U_{max}^{(2)}$  and  $U_{max}^{(3)}$  are the maximum allowable utilization indices of SDN controllers at the respective tiers.

Constraint (13) specifies that the average response time of the core, edge, and mobile controllers must not exceed a predefined threshold value established for the corresponding transaction. This condition applies to all controllers within the sets of available core network controllers, edge controllers, and mobile SDN controllers. The value of  $\tau_\varphi$  is determined to ensure adherence to specific Quality of Service (QoS) requirements. Constraints (14)–(16) pertain to the utilization index of each controller in the network, which must remain within the lower and upper utilization bounds.

These utilization limits are defined to maintain the system's overall QoS. The utilization index of each controller serves as a metric correlated with energy consumption, storage capacity, and data processing overhead. Furthermore, distinct utilization bounds were considered for controllers at different tiers, reflecting the varying operational capabilities inherent to each architectural level.

#### 4. Performance Evaluation of the SDN Controller Placement Method in Ultra-Dense Networks

To evaluate the performance of the developed SDN controller placement method for mobile nodes in ultra-dense networks, a simulation study was conducted. The NS-3 network simulator integrated with the Cloudsim framework was utilized as the simulation environment.

The evaluation accounted for the workloads of four heterogeneous application types:

**Category I (APPs(I)):** Applications comprising simple task workloads, such as basic web page processing.

**Category II (APPs(II)):** Image-based applications involving low-complexity image processing tasks.

**Category III (APPs(III)):** Basic video applications involving moderate workloads typical of simple video processing.

**Category IV (APPs(IV)):** Applications requiring the processing of complex data, such as 360-degree imagery and high-definition video..

As the classification progresses from Category I to Category IV, the tasks demand increasingly substantial resources, thereby reducing the probability of successful local execution. Consequently, higher-category tasks require significantly more energy and computational power than lower-category tasks.

To demonstrate the efficiency of the developed method, five system architectures were compared:

**System I:** A traditional multi-controller SDN employing multiple controllers in the core network without additional management schemes;

**System II:** A fog-computing-based SDN featuring a centralized topology with distributed edge SDN controllers integrated into fog nodes.

**System III:** An MEC-SDN-based network with a centralized topology where distributed edge SDN controllers are integrated into Multi-access Edge Computing (MEC) nodes.

**System IV:** A hybrid Fog-MEC SDN architecture utilizing distributed edge SDN controllers integrated into both fog and MEC nodes.

**System V:** A **Mobile SDN (MSDN)** network featuring mobile SDN controllers deployed on mobile nodes (e.g., public transit buses) supported by Fog-MEC computing units. This represents the proposed MSDN architecture.

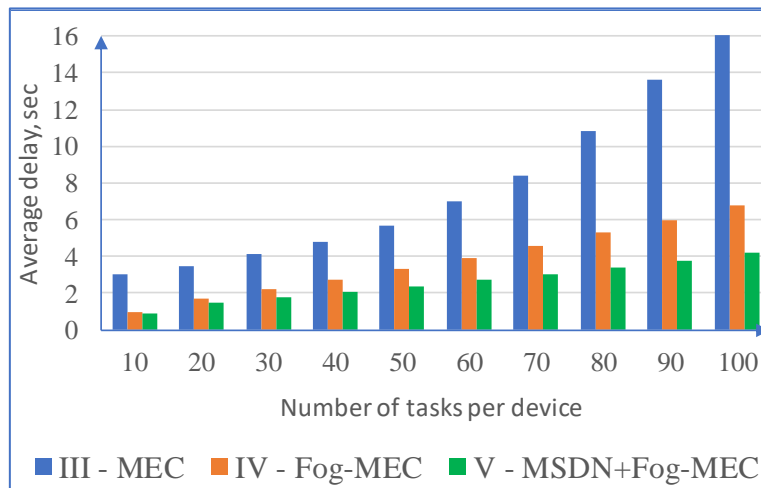
These systems were evaluated to highlight the effectiveness of each SDN deployment method relative

to the proposed approach. The primary performance metrics included energy consumption, latency, and availability..

Figure 3 illustrates the results for the average processing latency of computational tasks for the three MEC-oriented SDN configurations (Systems III through V). The simulation specifically considered tasks of low complexity, approximately corresponding to Category II applications.

In each scenario, a varying number of computational tasks were assigned to the mobile SDN end-nodes.

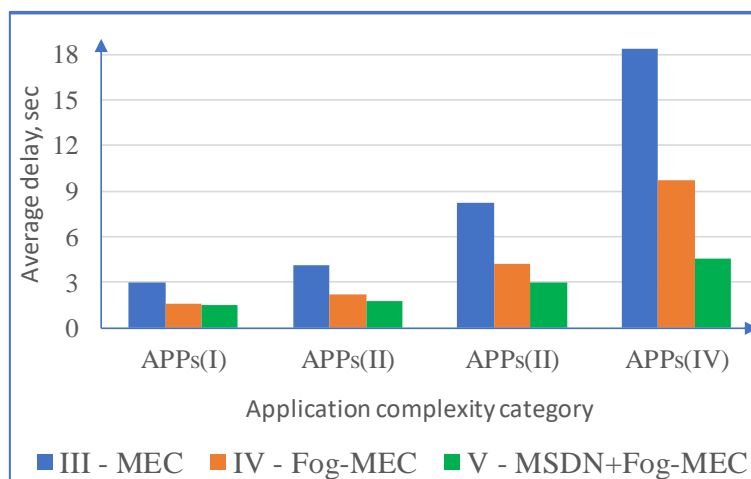
The average processing latency was analyzed as a function of the average task count. As the number of assigned tasks increased, the average latency rose across all three system types, primarily due to the overhead associated with managing a larger task volume. However, the proposed method outperformed all other configurations, achieving higher efficiency, which was particularly evident at higher task densities.



**Fig. 3.** Average latency as a function of the number of available low-complexity tasks

Figure 4 presents the relationship between average processing latency and task complexity. The average latency was calculated across the four previously defined

application categories. During this simulation, an average of 30 computational tasks were assigned to the mobile SDN end-nodes.



**Fig. 4.** Average latency across different application complexity categories

The simulation results indicate that the deployment of mobile SDN controllers significantly reduces the average time required to process computational tasks across various applications, with the most pronounced improvements observed for complex tasks with high workloads.

### Conclusions

This paper presents a method for deploying SDN controllers on mobile nodes within ultra-dense networks, representing a vital advancement in enhancing management efficiency for modern telecommunication systems. As part of the proposed methodology, a specialized MEC platform architecture was designed for SDN controller placement in 5G/6G ultra-dense environments, and a three-tier mobile multi-controller SDN model for ultra-dense sensor networks was developed.

The implementation of the proposed method demonstrates a significant performance advantage,

reducing transaction transmission latency by **60%** compared to traditional SDN-based edge computing models.

Furthermore, it achieves a reduction in energy consumption of up to 30%.

Future research will focus on developing an advanced load-balancing method among the controllers within the three-tier SDN architecture to further optimize resource distribution.

### Conflict of interest

The authors declare that they have no conflict of interest regarding this study, including financial, personal, authorship, or other, that could affect the study and its results presented in this article.

### Using artificial intelligence tools

The authors confirm that they did not use artificial intelligence technologies when creating the submitted work.

### СПИСОК ЛІТЕРАТУРИ

- Narwaria, A. and Mazumdar, A. P. (2023), "Software-Defined Wireless Sensor Network: A Comprehensive Survey", *Journal of Network and Computer Applications*, vol. 215, doi: <https://doi.org/10.1016/j.jnca.2023.103636>
- Kuchuk, H., Chumachenko, I., Marchenko, N., Kuchuk, N. and Lysytsia, D. (2025), "Method for calculating the number of IOT sensors in environmental monitoring systems", *Advanced Information Systems*, vol. 9, no. 3, pp. 66–72, doi: <https://doi.org/10.20998/2522-9052.2025.3.08>
- Khan, M.A., Rehmani, M.H. and Rachedi, A. (2022), "SDN-based gateway architecture for electromagnetic nano-networks", *Computer Communications*, vol. 184, pp. 160–173, doi: <https://doi.org/10.1016/j.comcom.2021.12.017>
- 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>
- Kuchuk, H., Husieva, Y., Novoselov, S., Lysytsia, D., 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>
- Cui, X., Gao, X., Ma, Y. and Wang, W. (2020), "A controller deployment scheme in 5G-IoT network based on SDN", *EURASIP Journal on Wireless Communications and Networking*, article number 248 (2020), doi: <https://doi.org/10.1186/s13638-020-01853-8>
- Kuchuk, H., Mozhaiev, O., Tiulieniev, S., Mozhaiev, M., Kuchuk, N., Lubentsov, A., Onishchenko, Yu., Gnusov, Yu., Brendel, O. and Roh, V. (2025), "Devising a method for energy-efficient control over a data transmission process across the mobile high-density Internet of Things", *Eastern European Journal of Enterprise Technologies*, vol. 4(4(136)), pp. 46–57, doi: <https://doi.org/10.15587/1729-4061.2025.336111>
- Panchenko, V., Kuchuk, H., Noskov, V., Leonov, S., Lipchanska, O. (2026), "Method of test pool synthesis for an intelligent high-density IoT edge-layer gateway", *Advanced Information Systems*, vol. 10, no. 1, pp. 50–57, doi: <https://doi.org/10.20998/2522-9052.2026.1.06>
- Kreutz, D., Ramos, F.M.V., Verissimo, P.E., Rothenberg, C.E., Azodolmolky, S. and Uhlig, S. (2015), "Software-Defined Networking: A Comprehensive Survey", *Proceedings of the IEEE*, vol. 103, is. 1, January 2015, pp. 14–76, doi: <https://doi.org/10.1109/JPROC.2014.2371999>
- Hu, F., Hao, Q., and Bao., K. (2014), "A Survey of Software-Defined Network and OpenFlow: From Concept to Implementation: From Concept to Implementation", *IEEE Communications Surveys & Tutorials*, pp. 2181–2206, doi: <https://doi.org/10.1109/COMST.2014.2326417>
- Ndiaye, M., Hancke, G. P., and Abu-Mahfouz, A. M. (2017), "Software Defined Networking for Improved Wireless Sensor Network Management: A Survey", *Sensors*, vol. 17(5), doi: <https://doi.org/10.3390/s17051031>
- Puente Fernández, J. A., García Villalba, L. J., and Kim, T.-H. (2018), "Software Defined Networks in Wireless Sensor Architectures", *Entropy*, vol. 20(4), doi: <https://doi.org/10.3390/e20040225>
- Miguel, M. L. F., Jamhour, E., Pellenz, M. E., and Penna, M. C. (2018), "SDN Architecture for 6LoWPAN Wireless Sensor Networks", *Sensors*, vol. 18(11), doi: <https://doi.org/10.3390/s18113738>
- Liao, Y., Wang, J. and Liu, J., 2020, A controller deployment scheme in 5G-IoT network based on SDN, *EURASIP Journal on Wireless Communications and Networking*, DOI: <https://doi.org/10.1186/s13638-020-01853-8>
- Khan, M.A., Rehmani, M.H. and Rachedi, A., 2022, SDN-based gateway architecture for electromagnetic nano-networks, *Computer Communications*, DOI: <https://doi.org/10.1016/j.comcom.2021.12.017>
- Al-Turjman, F., 2019, Intelligence and security in big 5G-oriented IoNT: An overview, *Future Generation Computer Systems*, DOI: <https://doi.org/10.1016/j.future.2018.08.028>

17. 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>
18. Kuchuk, H. and Malokhvii, E. (2024), "Integration of IoT with Cloud, Fog, and Edge Computing: A Review", *Advanced Information Systems*, vol. 8(2), pp. 65–78, doi: <https://doi.org/10.20998/2522-9052.2024.2.08>
19. Miguel, M. L. F., Jamhour, E., Pellenz, M. E., and Penna, M. C. (2018), "SDN Architecture for 6LoWPAN Wireless Sensor Networks", *Sensors*, vol. 18(11), DOI: <https://doi.org/10.3390/s18113738>

Received (Надійшла) 12.01.2026

Accepted for publication (Прийнята до друку) 15.04.2026

Publication date (Дата публікації) 22.05.2026

#### ВІДОМОСТІ ПРО АВТОРІВ/ ABOUT THE AUTHORS

- Шефер Олександр Віталійович** – доктор технічних наук, професор, завідувач кафедри автоматичної, 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](mailto:itm.ovshefer@nupp.edu.ua); ORCID Author ID: <https://orcid.org/0000-0002-3415-349X>;  
Scopus Author ID: <https://www.scopus.com/authid/detail.uri?authorId=57210203269>.
- Мигаль Станіслав Вікторович** – аспірант кафедри автоматичної, 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](mailto:stas.migal1998@gmail.com); ORCID Author ID: <https://orcid.org/0009-0007-4675-7358>.

#### Розробка методу побудови архітектури мобільної SDN для надшільних сенсорних мереж

О. В. Шефер, С. В. Мигаль

**Анотація. Актуальність.** Метод розміщення SDN-контролерів на мобільних вузлах надшільних мереж є важливою складовою підвищення ефективності управління сучасними телекомунікаційними системами. У контексті стрімкого зростання кількості підключених пристроїв і обсягів трафіку традиційні підходи до централізованого управління мережами виявляються недостатньо гнучкими та масштабованими. **Предметом** дослідження є методи побудови архітектури мобільної SDN для надшільних сенсорних мереж. **Метою** дослідження є розробка методу побудови архітектури програмно-визначеної надшільної сенсорної мережі, у якій деякі SDN-контролери встановлюються на мобільних вузлах граничного шару. **Отримані наступні результати.** Запропонована архітектура платформи MEC для методу розміщення SDN-контролерів на мобільних вузлах надшільних мереж 5G/6G. Розроблена тривірнева модель мобільної мультиконтролерної SDN для надшільної сенсорної мережі. Також розроблений метод розміщення SDN контролерів на мобільних вузлах надшільних мереж. **Висновок.** Використання запропонованого методу дозволяє зменшити затримку передачі транзакцій на 60% проти традиційних моделей граничних обчислень з урахуванням SDN, і навіть знизити споживану енергію на 72%.

**Ключові слова:** телекомунікаційна мережа, граничний шадозволяєр, мобільний пристрій, контролер, надшільна сенсорна мережа, OpenFlow-комутатор, стандарт 5G, мережа SDN.