UDC 621. 396.2 **doi: 10.26906/SUNZ.2024.1.206**

I. M. Syvolovskyi[1], V. P. Lysechko[1], O. S. Zhuchenko[1], O. M. Komar[2], V. V. P astushenko[1]

[1] Ukrainian State University of Railway Transport, Kharkiv, Ukraine
[2] National Aviation University, Kyiv, Ukraine

ANALYSIS OF METHODS FOR ORGANIZING DISTRIBUTED TELECOMMUNICATION SYSTEMS USING THE PARADIGM OF EDGE COMPUTING

Abstract: The article analyzes modern architectures of edge and fog computing systems, including OpenFog, F2c2C (Cloudlet), MELINDA, and architectures based on SDN and NFV. Particular attention is given to the study of Fog Computing from the conceptual and programmatic points of view. The advantages and limitations of the studied architectures in the context of IoT application are determined. Opportunities for enhancing telecommunication systems and improving the quality of service through the use of appropriate architectures are identified. The necessity of taking into account the specific needs and features of each system when choosing the appropriate fog computing architecture is proved. The need and relevance of further development and improvement of these architectures for optimal use are substantiated.

Keywords: N-tier architecture, edge computing, distributed systems, fog computing, Internet of Things, OpenFog, video stream processing, SDN, NFV, telecommunication system.

Abbreviations

IoT is an Internet of Things;
F2c2C is a Fog-to-cloudlet-to-Cloud;
MELINDA is a Multilevel Information Distributed Processing Architecture;
MLT is a Measurement Level Task;
FLT is a Feature Level Task;
DLT is a Decision Level Task;
SDN is a Software-Defined Network;
NFV is a Network Functions Virtualization;
SDNFV is a Software-Defined NFV
QoS is a Quality of Service.
ASTP is an Adaptive Selection and Task Priority.
SuVMF is a Software-defined Unified Virtual Monitoring Function.

Problem statement

Modern telecommunication systems that process large amounts of data and require minimal latency face the need to implement specialized architectures that allow for optimal resource utilization, improve service quality, and reduce delays. In particular, there is a need to research and implement Fog Computing architectures that facilitate efficient operation and reduce data transmission delays. These architectures also allow for the specific needs of different IoT systems and applications, making them more adaptable and productive. The study was necessitated by the need to optimize telecommunication systems in response to modern requirements and the growing amount of data in the IoT field [1, 2].

Analysis of recent research and publications. Despite the relevance of the study of edge computing architectures, there is currently no systematization in this area, no comprehensive analysis and comparative evaluation of different architectures, although many authors, both domestic and foreign, have partially studied this issue [1–11]. Given the rapid development of this area and its potential for the introduction of new technologies, the availability of such a study is a scientific and practical need.

The purpose of the article is to analyze potential solutions for improving distributed telecommunication systems using IoT and edge computing, as well as to justify the need to select optimal solutions for specific challenges and needs.

Presentation of the main material

In recent years, the availability of cloud technologies has led to the widespread integration of Cloud Computing into server systems, which has radically changed the paradigm of infrastructure and computing environments in business and technology fields, including telecommunications. The popularization of this paradigm and its widespread adoption was primarily due to the virtually unlimited expansion of server system resources through virtualization of all components.

In terms of architectural solutions, in the context of cloud-based server systems, the popularity of cloud integration has led to specific patterns or architectural solutions being used to organize these systems. One of these patterns is a two-tier architecture, which involves dividing the server system into two tiers: frontend and backend. The frontend is responsible for processing user requests and interacting with them, while the backend performs data operations and computations. This two-tiered architecture is a common approach that provides a certain level of standardization and allows for effective separation of functional responsibilities and scalability of the system (Fig. 1).

An architecture such as the one shown in Figure 1 is acceptable in the context of conventional client-server applications which a user interacts. But, in the context of IoT, tasks with the following parameters may arise.

1. Network bandwidth. A large number of connected IoT devices that constantly generate data can create significant problems with cloud network bandwidth, lead to overload and reduce the quality of service.

2. Latency. A significant distance between the IoT device and the cloud server can create a delay in data exchange. This can become a critical problem in cases requiring quick reactions, such as security systems or medical devices.

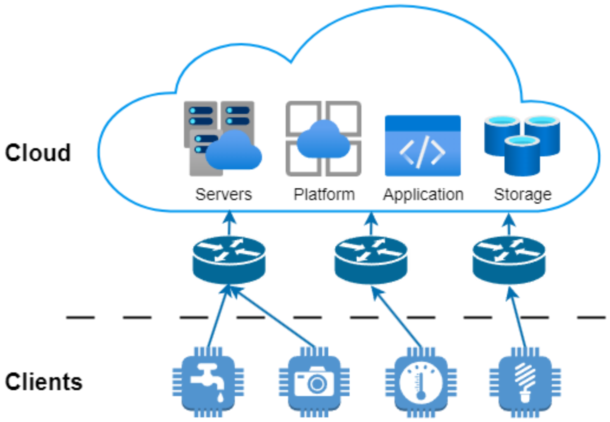

Fig. 1. Two-tier Cloud Computing architecture

3. Data security. The increased number of connected devices creates opportunities for cyberattacks and data breaches.

4. Scalability and administration. With a large number of connected devices, there are problems with administration and management.

5. Ensuring the viability of devices. A number of IoT devices have limited resources, such as batteries. Ensuring the longevity of the Internet of Things and their reliability requires the development of effective strategies for managing energy consumption and monitoring device health.

6. Interoperability and standardization. Different manufacturers may use different protocols to connect their devices to the IoT network, which can affect compatibility and integration between devices and systems.

To solve these problems, the Edge Computing paradigm was developed with the purpose of transferring part of the computing (functionality) to nodes that are closer to the devices than the cloud. At the same time, a computing node can be not only a data center, but any device with computing capabilities.

The emergence of Edge Computing set the general concept of such systems, which contributed to the development of new architectures later.

Fog Computing is an architecture concept in which an additional layer of processing nodes is added between the cloud layer and the device layer. It is often mentioned as a synonym for Edge Computing and is considered depending on the interpretation: both as an additional layer of the cloud layer and as an additional layer next to devices.

Dew Computing is a microservice concept that is embodied in a platform where devices can interact with each other continuously within a single "local" network, and this interaction takes place without sending data to cloud resources. Smart devices are one example of how this concept can be used.

Fog-Dew Computing is a synthesis of aforementioned architectures that utilizes its main advantages: devices operate as autonomous devices that do not require a constant connection to the Internet (to the cloud), but are connected to a local server. However, the local server interacts with cloud resources and is responsible for providing services to devices.

Nowadays, according to Google Scholar, Scopus, and Web of Science statistics, Fog Computing attracts the most attention from researchers due to its versatility and applicability in various areas of the Internet of Things (IoT) [3].

It should be noted that the concept of Dew Computing is limited, with opportunities for usage only in certain distributed telecommunication systems. Therefore, further research should focus on Fog Computing architectures.

When analyzing architectures, it is important to adhere to a clear definition of the concepts. It is essential to distinguish between the concepts of layer and tier, which are used synonymously in practice, but have a significant difference. A layer is a way of logically structuring the components of a software solution, while a tier is a way of physically structuring the infrastructure [4].

OpenFog N-tier architecture

In 2017, Princeton University and some leading IT companies from various industries formed the OpenFog consortium. The result of the collaboration was the OpenFog Reference Architecture document, which was the basis for the IEEE 1934-2018 protocol (Institute of Electrical and Electronics Engineers) [5].

This document is a new model of service architecture - FaaS (Fog as a Service), which includes the previously known ones: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS), but with adaptation to Fog Computing, as well as ways to expand them.

In comparison to other architectures, OpenFog defines its advantages by using the SCALE acronym, which stands for.

1. Security - additional ways to achieve data security.

2. Cognition - ensuring autonomy by understanding the goals of the system's clients.

3. Agility - ensuring fast and affordable scaling.

4. Latency - reducing latency to ensure real-time processing.

5. Efficiency - dynamic allocation of system resources to achieve maximum efficiency.

The architecture is based on 8 basic principles called «pillars»: Security, Scalability, Openness, Autonomy, Programmability, RAS (Reliability, Availability, Serviceability), Agility, and Hierarchy. The description of each pillar is a set of recommendations and requirements for the system.

Fig. 2 demonstrates the OpenFog architecture with one level of Fog nodes.

According to Fig. 2, the OpenFog architecture does not impose any strict limitations on the number of layers. It makes possible to adapt the structure of the architecture to a specific subject area, so the number of cloud node tiers is arbitrary, and the presence of cloud and fog layers is optional.

Currently, despite the high level of standardization and description of this architecture, it is rarely used in practice due to the lack of a clear focus on a particular industry.

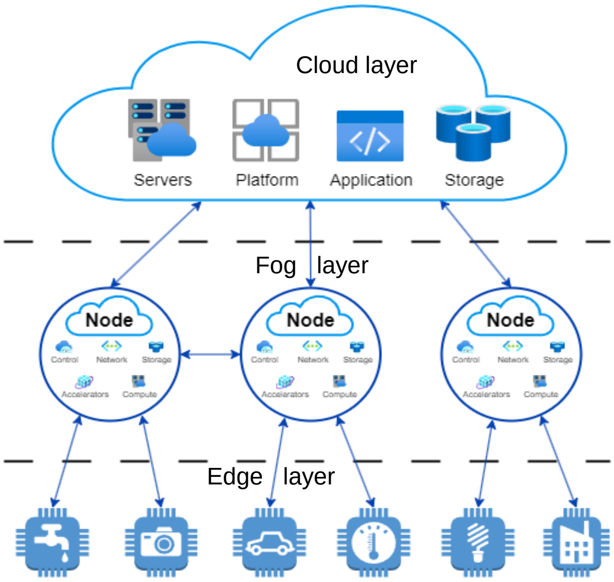

Fig. 2. N-tier OpenFog architecture

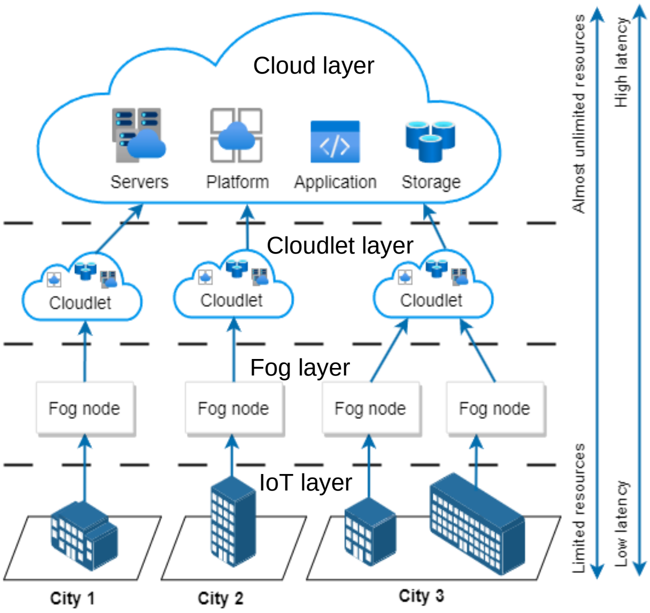

Fig. 3. Graphical representation of the F2c2C (Cloudlet) architecture on the example of the Smart City system

As a result, the main focus of current scientific and practical research is on Fog Computing architectures that are more adapted to specific domains and industries.

Cloudlets

The research of the OpenFog architecture has proven that one of the main challenges in the development of Fog Computing systems is to determine the optimal number of Fog node levels, their location, and allocation of resources for their operation.

The Smart City industry has traditionally used a centralized approach to organizing systems and data using Cloud Computing technologies. However, issues such as data protection, increased latency requirements, and energy efficiency have led to the consideration of decentralized architectures.

Some industries that are heavily utilizing the Internet of Things (IoT), primarily telecommunications systems, have the potential to develop distributed architectures at the conceptual level. For example, in the field of Smart City, researchers have introduced an innovation by proposing an architectural approach that offers territorial division (Fig. 3) [6].

Three conceptual levels are introduced to divide the system architecture by territorial basis: micro (building level), meso (neighborhood level), and macro (city or cloud level).

Also, in the context of this architecture, the concept of «cloudlet» is introduced, which implies a small data center that is located as close as possible to potential customers, unlike the cloud.

Since this architecture focuses on data management, three types of data are introduced according to their age.

1. Real-time data: produced by devices and nodes at the micro and meso levels in places where minimal latency is required.
2. Latest data: generated at the meso level, the result of real-time data processing.
3. Historical data: data stored in the cloud (macro level).

Guided by these concepts, one can define three main layers of this architecture (except for the device layer):

1. Fog layer: is as close to the devices as possible, works with real-time data, and located at the micro and meso levels.
2. Cloutlet-node layer: an intermediate layer between the cloud and fog, located in the same city as the devices (macro level), performs tasks of processing the «closest» data.
3. Cloud layer: processes and stores historical data, has unlimited resources.

Thus, this architecture combines the advantages of centralized and decentralized architectures: operation in a heterogeneous IoT environment, low load on the cloud network, the ability to process critical data in real time, etc.

MELINDA

One of the most complex systems in the telecommunications industry is video monitoring systems with real-time object detection. The traditional approach using Cloud Computing involves transferring the raw video stream to cloud data centers, where it is processed and then transmitted to the client. This approach has serious infrastructure-related drawbacks, such as.

1. High network saturation (each Full-HD camera generates a video stream of up to 12 Mbps), which creates problems with scaling the system in the form of limited cloud network bandwidth.
2. High and unstable latency when transferring data from the client to the cloud.
3. High resource and power consumption caused by the need to store a large amount of low-value data (video stream frames without objects or without changes).

To solve these problems, Neto A.R. proposed a three-tier Fog Computing architecture (Fig. 4) [7].

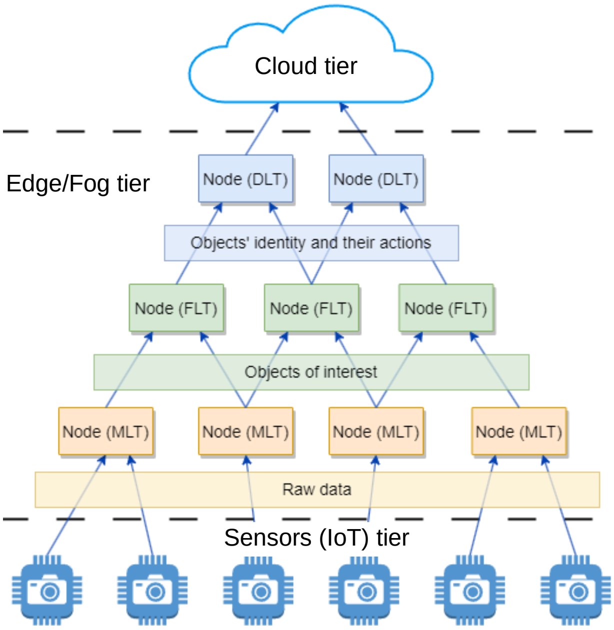

Fig. 4. Three-tier Fog Computing architecture optimized for video stream processing

According to Figure 4, the basic principle of this architecture is to reduce the amount of data by processing the video stream in multiple stages. Video stream processing consists of three steps.

– filtering the video stream to select only those frames that may contain an object;

– object identification;

– interpretation of the object's appearance (linking it to a specific event) for making decisions later.

Respectively, the architecture contains three types of processing nodes that perform tasks of different levels: Measurement Level Task (MLT), Feature Level Task (FLT), and Decision Level Task (DLT).

At the same time, this infrastructure is supported by the MELINDA (Multilevel Information Distributed Processing Architecture) software architecture, which consists of two subsystems.

1. Processing Subsystem, which consists of nodes for monitoring, resource allocation, and processing.

2. Management Subsystem, which consists of nodes for processing end-user requests (data extraction) and components for high-level system monitoring.

The Data Communication Manager component, common to both subsystems, is used for communication. It is assumed that each of the components is located on a separate node, and there can be several components of the same type.

SDN/NFV

The above architectures consider a Fog Computing system from a conceptual and programmatic point of view, following traditional methods of networking, using tree-like structures of Ethernet routers and mobile base stations. Many recent concepts for IoT include the use of new technologies: 5G as a data transmission technology, and SDN (Software Defined Network) and NFV (Network Functions Virtualization) for organizing network interaction [8].

The use of 5G is aimed at reducing the latency, cost, and power consumption of devices, as well as introducing new types of IoT systems that were previously impossible or difficult to implement.

Instead, the use of SDN and NFV enables us to look at some aspects of a Fog Computing system from a different perspective. Figure 5 shows the architecture of a system using Software-Defined NFV on a layer with Fog nodes.

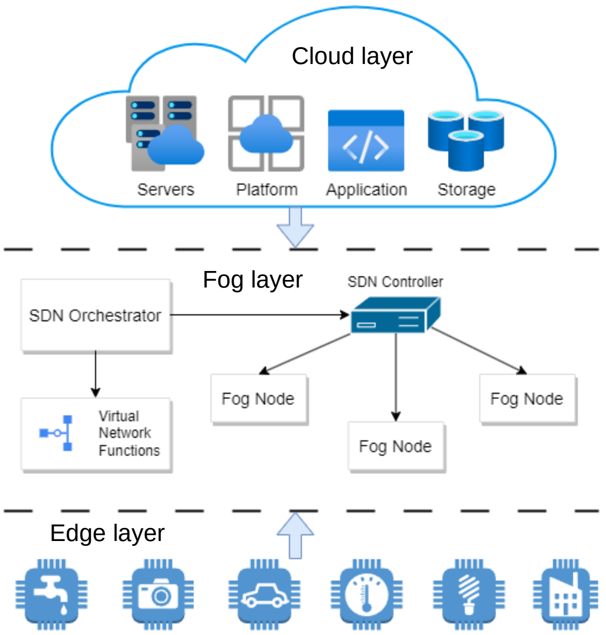

Fig. 5. System architecture using SDNFV on a layer with Fog nodes

Fig. 5 depicts that the basis of the fog layer is the SDN controller, which is responsible for processing all network traffic. It is assumed that this controller is an intermediate link between processing nodes and clients, and it is guided by 4 types of actions, namely: creating, modifying, executing, and terminating a task.

Thus, when a client requests a task from the orchestrator, it can check the validity of this request and request a certain amount of resources to execute this task.

In addition, it should be noted that the system takes into account five indices when executing tasks.

1. Cost index - the distance from the source to the node. The smaller the distance, the better the architecture.

2. Time index - the sum of the time spent on data transfer and the time of task execution. The smaller, the better.

3. Throughput index - the uniformity of user distribution in relation to traffic.

4. Energy Consumption index - the consumed energy, taking into account the energy consumption for system idle time. The lower, the better.

5. Capacity of machines in the cloud layer.

In order to test the performance of this architecture, a modeling environment consisting of MATLAB software and the EstiNet simulator was created. The main parameters characterizing the quality of the system were measured: total delay time, percentage of successfully completed tasks (reliability), and task processing speed (Quality of Service).

To analyze the performance of the proposed architecture, two existing similar architectures were selected, which are also based on SDN: ASTP (Adaptive Selection and Task Priority) and SuVMF (Software-defined Unified Virtual Monitoring Function) [9, 10].

The measurement results showed that the aforementioned architecture has higher values of the examined indicators compared to similar architectures: 90% reliability rate (vs. 85% in ASTP and 70% in SuVMF); 90% QoS (vs. 82% in ASTP and 68% in SuVMF). The results show that the use of SDN as a means of load balancing between resources can significantly improve system performance, which is highly relevant in various areas of IoT related to real-time processing, such as Industrial IoT [11]. Thus, further research on the use of this technology in IoT and fog computing systems is relevant.

Conclusion

The article investigates the methods of organizing distributed telecommunication systems using the Internet of Things (IoT), as well as the existing paradigms of Edge and Fog Computing.

During the analysis, the following fog computing architectures were reviewed: OpenFog, F2c2C (Cloudlet), MELINDA, and SDNFV architecture.

According to the results of the analysis, we can conclude that the researched architectures show the ability to solve the main problems of cloud architectures: high latency and high network saturation, by transferring part of the computation to nodes at the edge of the network.

In the field of distributed telecommunication systems, these architectures provide prospects for further development of performance improvement methods.

However, in practice, it is important to consider the specific requirements and features of each system when choosing a suitable fog computing architecture. In addition, the development and improvement of these architectures is an essential task, as they must be optimized for various applications, scenarios, and industries.

References

1. Lysechko V., Zorina O., Sadovnykov B., Cherneva G., Pastushenko V.: Experimental study of optimized face recognition algorithms for resource – constrained. Academic journal: Mechanics Transport Communications, 2023, vol. 21, issue 1, article №2343, ISSN 2367-6620/
2. Lysechko V., Syvolovskyi I., Shevchenko B., Nikitska A., Cherneva G.: Research of modern NoSQL databases to simplify the process of their design. Academic journal: Mechanics Transport Communications, 2023, vol. 21, issue 2, article №2363, ISSN 2367-6620/
3. Naha R.K., Garg S., Georgekopolous D., Jayaraman P.P., Gao L., Xiang Y., Ranjan R.: Fog Computing: Survey of Trends, Architectures, Requirements, and Research Directions, IEEE Access, vol. 6, pp. 47980-48009, 2018.
4. Fowler M. Patterns of Enterprise Application Architecture. 1st Edition. – Addison-Wesley Professional, 2002. – 560 P. – ISBN 978-0321127426.
5. IEEE, "Ieee standard for adoption of openfog reference architecture for fog computing," aug 2018, standard No. 1934-2018, Active since Aug. 2nd., 2018. [Online]. Available: https://standards.ieee.org/standard/1934-2018.html
6. Sinaeepourfard A., Krogstie J., Petersen S. A., Ahlers D.: F2c2C-DM: A Fog-to-cloudlet-to-Cloud, 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), pp. 590–595, 2019.
7. Neto A.R.: Edge-distributed Stream Processing for Video Analytics in Smart City Applications, 2021, 10.13140/RG.2.2.10968.57604.
8. Sreekanth G.R., Ahmed Najat Ahmed S., Sarac M., Strumberger I., Bacanin N., Zivkovic M.: Mobile Fog Computing by Using SDN/NFV on 5G Edge Nodes, Computer Systems Science and Engineering, vol. 41, pp. 751-765, 2021.
9. Wang J., Li D., Adaptive computing optimization in software-defined network-based industrial internet of things with fog computing, Sensors, vol. 18, no. 8, pp. 1–14, 2018.
10. Choi T., Kang S., Yoon S., Yang S., Song S. et al. «SuVMF: Software-defined unified virtual monitoring function for SDN-based large-scale networks» in Proc. CFI, ACM, Tokyo, Japan, pp. 1–6, 2014.
11. Alam M., Ahmed N., Matam R., Mukherjee M., Barbhuiya F.A.: SDN-Based Reconfigurable Edge Network Architecture for Industrial Internet of Things, IEEE Internet of Things Journal, 2023.

Received (Надійшла) 18.11.2023
Accepted for publication (Прийнята до друку) 31.01.2024

АНАЛІЗ МЕТОДІВ ОРГАНІЗАЦІЇ РОЗПОДІЛЕНИХ ТЕЛЕКОМУНІКАЦІЙНИХ СИСТЕМ З ВИКОРИСТАННЯМ ПАРАДИГМИ ГРАНИЧНИХ ОБЧИСЛЕНЬ

І. М. Сиволовський, В. П. Лисечко, О. С. Жученко, О. М. Комар, В. В. Пастушенко

Анотація: В статті проаналізовано сучасні архітектури систем граничних та туманних обчислень, включаючи OpenFog, F2c2C (Cloudlet), MELINDA та архітектуру з використанням SDN та NFV. Особливу увагу приділено дослідженню Fog Computing з концептуальної і програмної точок зору. Визначено переваги та обмеження досліджених архітектур у контексті застосування в IoT. Виявлено можливості для вдосконалення телекомунікаційних систем і покращення якості обслуговування через використання відповідних архітектур. Доведено необхідність врахування конкретних потреб і особливостей кожної системи при виборі відповідної архітектури туманних обчислень. Обгрунтовано необхідність та актуальність подальшого розвитку та вдосконалення цих архітектур для оптимального використання.

Ключові слова: N-рівнева архітектура, граничні обчислення, розподілені системи, туманні обчислення, інтернет речей, OpenFog, Fog Computing, обробка відеопотоку, SDN, NFV, телекомунікаційна система.