

Migration routing algorithm for microservice based Fog computing system

Ekaterina Kuzmina
Department of Telecommunication
Networks and Data Transmission,
Saint-Petersburg State University of
Telecommunications
Saint Petersburg, Russia
kuzmina120601@yandex.ru

Meriem Tefikova
Department of Telecommunication
Networks and Data Transmission,
Saint-Petersburg State University of
Telecommunications
Saint Petersburg, Russia
tmrvmr@mail.ru

Artem Volkov
Department of Telecommunication
Networks and Data Transmission,
Saint-Petersburg State University of
Telecommunications
Saint Petersburg, Russia
artemanv.work@gmail.com

Ammar Muthanna
Department of Telecommunication
Networks and Data Transmission,
Saint-Petersburg State University of
Telecommunications
Saint Petersburg, Russia
muthanna.asa@spbgut.ru

Andrey Koucheryavy
Department of Telecommunication
Networks and Data Transmission,
Saint-Petersburg State University of
Telecommunications
Saint Petersburg, Russia
akouch@mail.ru

Abstract— The new generation of communication networks has provided the basis for the realization of new classes of services, one of which are URLLC services. These services include concepts such as the Tactile Internet. The transmission medium for URLLC requires high bandwidth and reliability of communication channels as well as ultra-low latency in the network. To cover these requirements, various technologies such as fog computing and microservice architecture are being introduced into the network architecture. The latter is more often recommended together with support for migration of microservices between devices. This paper considers one of the most important migration tasks, namely determining an efficient path to the destination server. The parameters that need to be considered for efficient routing in microservice migration were identified, as well as the network architecture layer that should control the migration. The results show that the proposed solution is preferred for application over the AODV protocol.

Keywords— *Fog computing; 6G; Protocol; service migration*

I. INTRODUCTION

Previously, communication networks could not meet all the requirements strictly set for new approaches of machine-to-machine communication, but now there is a qualitative leap in the capabilities of modern networks. The most recently released mobile communication standard is IMT-2020 (5G), which has greatly surpassed its predecessors in all directions. Data transmission speed, the number of devices connected to the network, network reliability - the fifth generation of communication networks provided growth of all these indicators. It is also important that the network latency has been significantly reduced [1]. All this allowed us to approach the realization of such concepts as the Tactile Internet. However, the requirements of the Tactile Internet to the transmission medium are very specific, for example, the network latency should not exceed 1 ms [2]. Moreover, in addition to the changing composition of the technological background of human life, its scale is also changing. The

number of devices connected to the network is growing relentlessly, and at the same time the amount of traffic transmitted over the network is also increasing [3]. Classical approaches to the construction of communication networks will simply cease to cope with the service of such a large number of requests, which entails the impossibility of providing the necessary QoS. Nevertheless, researchers have already proposed a huge number of innovative solutions to avoid the worst-case scenario of communication networks development. These include the concept of Fog computing and microservice architecture. However, the implementation of new solutions in the upper layers of the network infrastructure requires partial or complete changes in the technologies used at the lower layers. Thus, for quality operation of a network containing nodes supporting microservice architecture, it is necessary to implement the migration of microservices between network devices, and this process necessitates the development of new routing principles, when considered in relation to different network topologies and architectures.

II. THE AODV ROUTING PROTOCOL

AODV (Ad hoc On-Demand Distance Vector) is an open-source reactive routing protocol which is used in wireless self-organizing networks [4]. It is designed for mobile ad-hoc MANETs, Sensor networks and other wireless networks. AODV creates and maintains dynamic routes between nodes.

The working principle of the protocol can be described as follows [5]:

1. When there is a need to transmit data, the sending node starts searching for a route to the destination node. It sends a RREQ (Route Request) message to all its neighboring nodes.
2. Neighboring nodes forward the RREQ message to their own neighbors, and this continues until the message

reaches a node that knows the route to the destination node, or has a record of a shorter route to it.

3. When a RREQ message reaches either the destination node or a node that knows the route to it, the corresponding node sends a RREP (Route Reply) message back to the source node, the initiator of the route search. The RREP message follows the path specified in the RREQ message. In the process of returning RREP message, each node encountered on the path remembers the route to the sending node. After receiving the RREP message, the source node knows the path to the destination node.

One of the main advantages of the AODV protocol is its efficiency in the use of network resources: since routes are established only, when necessary, a constant stream of service messages is not transmitted over the network, as in the work of proactive routing protocols. This allows you to reduce the load on the network and reduce the requirements for the resources of nodes. Moreover, the results of the protocol determine the shortest route to the destination node, which means that data transmission will require fewer transitions through transit nodes. AODV also has the property of self-organization, which allows network nodes to dynamically adapt to changes in the network.

AODV protocol is preferred and applicable to solve the problem of microservice migration route determination in Fog computing network, but it does not allow to consider all the necessary circumstances of path selection to the destination node, so there is a need to create a new solution.

III. DEFINING THE PARAMETERS

The AODV protocol involves choosing the shortest route to the destination node, and the shortest route is determined by the number of hops required to reach the destination. The mentioned indicator is more often referred to as Hop-count. One hop implies passing through one network node, in other words, one transit section on the packet's path to the destination.

However, in the context of the task of route selection for microservice migration in the considered dynamic Fog computing network, it is required to take into account more network state parameters than the AODV protocol algorithm allows. This is due to the fact that the nodes in the network are represented by Fog devices performing different computational tasks, respectively spending their computational resources. It is necessary to balance the distribution of resources in the network, so it is necessary to choose a route whose transit nodes do not perform any resource-intensive tasks at the time of migration. For this purpose, the main server only needs to have information about the CPU and RAM utilization of each node in the network.

In addition, the link state indicators between the network nodes are also important for solving the described problem. Migration must be performed quickly and reliably enough, which requires from the communication channel a sufficiently low delay, and packet losses must also be minimal, otherwise the time spent on migration will grow.

Thus, the most important network state parameters for determining the microservice migration route in the current task are:

- CPU load on network nodes;
- RAM load on network nodes;
- Delay on communication channels between nodes;
- Share of packet loss on communication channels between nodes.

By tracking these parameters, it is possible to find a migration path that will be the most efficient, with the lowest latency, the lowest proportion of lost packets, and the most reliable due to the fact that the transit nodes, which covers the route, will have enough resources to pass the microservice through them, or their load will not approach the limit in contrast to the rest of the nodes in the network, in the process of migration to the destination.

The described characteristics should be collected in the form of real-time and continuous statistics for Fog zones. There should be a temporary or permanent data repository where the collected statistics will be recorded so that, if necessary, it will be possible to analyze the load in the zone over a certain period of time. This is also necessary for predicting the loads that will occur in Fog zones in the future.

IV. PROPOSED ALGORITHM TO ROUTE SELECTING

Turning to graph theory, it can be seen that a route has an important generalizing characteristic - the path cost, also referred to as Cost. The definition of a migration route should be based on the path cost, which will be calculated on the basis of the described considered parameters.

The routing tables of the network nodes can be represented by a graph, and then path finding algorithms in the graph can be used to find a path. One of the classical algorithms for solving such a problem is Dijkstra's algorithm. The algorithm assumes that each edge of the graph has its own weight, and the progression to the final node in the graph should follow the edges with the lowest weights among the possible neighbors of the origin node and transit nodes.

In the microservice migration pathfinding problem, there is no need to search for paths from the initial node of the graph to all other nodes, since the final destination is known in advance, it is determined by existing algorithms. Therefore, it is necessary to find only one path, which is the "shortest" path to the destination node. The "shortest" path is the route whose cost is minimal among the other available ones.

However, the costs of edges are not known in advance. They should be calculated and then the algorithm should be started. In general, the considered Fog computing network can be represented by the graph shown in Figure 1:

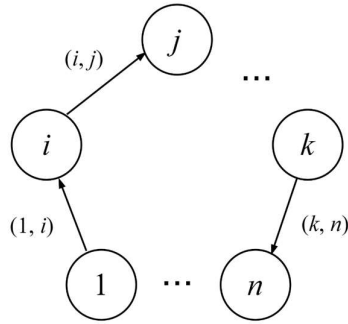


Figure 1. Generalized graph of the topology of the considered network

Figure 1 shows the generalized topology of the considered network. Let node 1 be the network node from which the microservice should be migrated, and node n be the network node to which the microservice should migrate. The graph is directed, since the microservice must exactly pass each node only once during the migration process, in other words, no loops in the routes should be formed.

Then, to determine the shortest route from node 1 to node n, we need to find all the weights of the edges encountered on the path to the destination node. The cost of an edge in this case should take into account not only the characteristics of the edge itself, but also the characteristics of the node, as was clarified earlier. Then the cost of an edge or link (i,j) will be determined as follows:

$$f_{(i,j)} = [\text{edgeOptions}]_{(i,j)} + [\text{topOptions}]_j, (1)$$

Where $[\text{edgeOptions}]_{(i,j)}$ - parameters describing the state of the edge, and $[\text{topOptions}]_j$ - parameters describing the state of the vertex to which the edge leads. The mentioned parameters were previously defined as CPU load of node j, RAM load of node j, delays in communication channels (i,j), PacketLoss in communication channels (i,j). All these characteristics have different units of measurement and different priorities, so they need to be brought to an equivalent form later.

Based on expression 1, the cost of the entire route can be expressed as:

$$F_x(1,n) = \sum_{i,j} f_{(i,j)}, (2)$$

and the required migration path is defined as:

$$\min \{ F_x(1,n) \}. (3)$$

Returning to the parameters describing the state of the network, as already mentioned, it is necessary to take into account that each of the selected parameters has its own magnitude of influence on the result and its own units of measurement. It is necessary to determine the importance of each parameter in relation to the others, since not all of them equally influence the choice of one or another path to the destination device. The importance of a parameter in this context is called its weight.

Parameter weights should be determined using methods of comparing a set of objects with each other based on the conditions of the task, such approaches include the Method of Pairwise Comparisons.

As a result of the calculations, the CPU, Delay, and PacketLoss parameters have weights of 0.275 and RAM has a weight of 0.175.

Thus, the target function from expression 1 of finding the cost of one edge (i,j) of the graph, with the found weights and certain parameters of the network state will be of the form:

$$f_{(i,j)} = [\text{CPU}]_j \cdot w_{\text{CPU}} + [\text{RAM}]_j \cdot w_{\text{RAM}} + D_{(i,j)} \cdot w_D + [\text{PL}]_{(i,j)} \cdot w_{\text{PL}}, (4)$$

$$w_{\text{CPU}} = w_D = w_{\text{PL}} = 0,275, w_{\text{RAM}} = 0,175.$$

Figure 2 shows the proposed microservice migration algorithm for route selection.

It is worth noting that the layer of the network architecture that controls this process (Figure 2) should be the Fog Computing Orchestrator. This task falls under the list of reasons for its use and implementation, because it is the orchestrator that deals with load distribution between Fog nodes, forecasting, prioritization of various tasks performed by Fog nodes, etc.

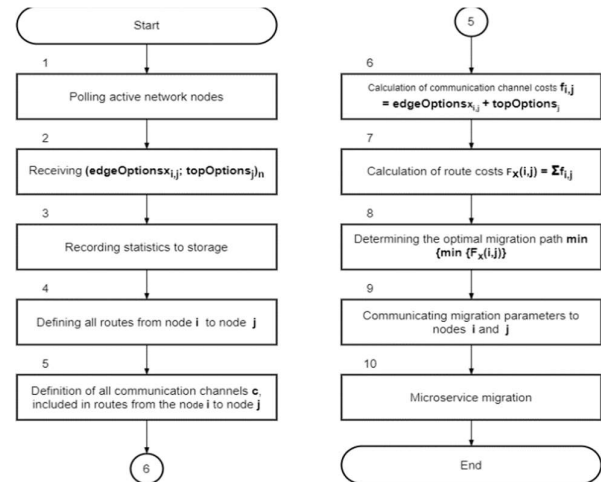


Figure 2. Proposed migration routing algorithm for microservice architecture

V. RESULTS

To test the performance of the proposed migration route determination algorithm and to evaluate its performance, 5 experiments were conducted on a model-based Fog computing network.

To conduct the experiments, the nodes of the model network were started and put into operation. Within the framework of these experiments, the nodes were subjected to artificial increase and decrease of load in order to collect

objective statistics on the performance of two routing algorithms: the classical route selection algorithm of the AODV protocol, which is based on the Hop-count metric, i.e., evaluates paths according to the choice of the shortest route by the number of transitions through transit nodes to the destination server, as well as the proposed algorithm for determining the migration path, which is based on the search for a route on less loaded links connecting and less loaded links.

Summary graphs on the indicators of network nodes CPU, RAM, Delay in data transmission (Delay), packet loss (PacketLoss) when using the mentioned algorithms for 5 experiments are presented in Figure 3.

The results show that the proposed algorithm selects a route less prone to loss of transmitted packets, the superiority over the AODV protocol algorithm can be estimated at 7% in this indicator. Also, the proposed algorithm considers the load at each node of the route, which is confirmed by 77% and 14% reduction in CPU and RAM resources consumed at the transit nodes of the network, respectively. At the same time, the delay performance increased by 4%, which is explained by the fact that previously the CPU and Delay parameters were determined to be equivalent in the considered problem, and the network nodes were subjected to high loads of 90% CPU utilization, with the link delay not taking such high values in the measurements, respectively, the differences in the parameters were large, in which case CPU caused a strong increase in the cost of the whole route. However, the algorithm allows us to adjust the weights of the parameters and the list of parameters themselves as needed for a particular task.

The experimentally proven efficiency of the algorithm gives an understanding that when performing microservice migration in a dynamic Fog computing network, it is the proposed algorithm that should be used to select the microservice migration route. In other words, it is preferred for solving this problem because it shows less utilization of network resources when migrating a microservice along the route that the algorithm determines, which means that the proposed solution takes into account the peculiarities of the considered network and improves its performance.

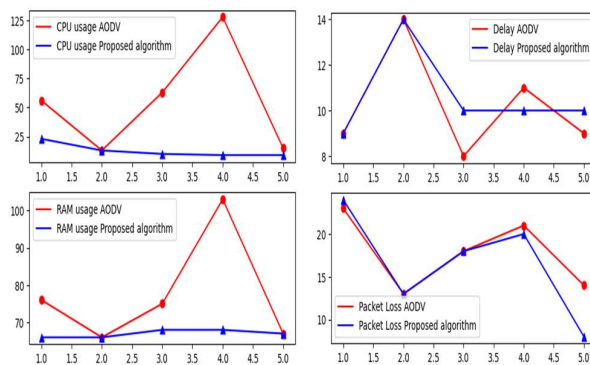


Figure 3. Load indicators when using the two algorithms

VI. CONCLUSION

In this paper, an algorithm for selecting the migration route of a microservice in a Fog computing network has been

proposed. The architecture layer that controls the migration process has been defined. The developed solution has been tested together with the already classical approach used by the AODV protocol, and from the results obtained it is clear that the proposed algorithm is more efficient than the AODV protocol algorithm for solving the microservice migration path finding problem. By improving the network performance that is observed in the microservice migration process using the proposed algorithm, it is possible to achieve minimization of network failures and provide the required QoS, which is an undeniably important issue in the deployment of fifth generation communication networks.

ACKNOWLEDGMENT

The studies at St. Petersburg State University of Telecommunications. Prof. M.A. Bonch-Bruевич was supported by the Ministry of Science and High Education of the Russian Federation by grant number 075-15-2022-1137.

REFERENCES

- [1] Recommendation ITU-R M.2083-0: IMT Vision, "Framework and overall objectives of the future development of IMT for 2020 and beyond," Sep. 2015.
- [2] D. Calandra, F. G. Praticò, A. Cannavò, C. Casetti, and F. Lamberti, "Digital twin- and extended reality-based telepresence for collaborative robot programming in the 6G perspective," *Digit. Commun. Netw.*, 2022.
- [3] A. A. Ateya, A. Muthanna, A. Koucheryavy, Y. Maleh, and A. A. El-Latif, "Energy efficient offloading scheme for MEC-based augmented reality system," *Cluster Comput.*, vol. 26, no. 1, pp. 789–806, 2023.
- [4] D. M. Hilty et al., "A review of telepresence, virtual reality, and augmented reality applied to clinical care," *J. Technol. Behav. Sci.*, vol. 5, no. 2, pp. 178–205, 2020.
- [5] M. Osama, A. A. Ateya, S. Ahmed Elsaid, and A. Muthanna, "Ultra-reliable low-latency communications: Unmanned aerial vehicles assisted systems," *Information (Basel)*, vol. 13, no. 9, p. 430, 2022.
- [6] V. Benavente, L. Yantas, I. Moscol, C. Rodriguez, R. Inquilla, and Y. Pomachagua, "Comparative analysis of microservices and monolithic architecture," in *2022 14th International Conference on Computational Intelligence and Communication Networks (CICN)*, 2022, pp. 177–184.
- [7] A. Razzaq and S. A. K. Ghayur, "A systematic mapping study: The new age of software architecture from monolithic to microservice architecture—awareness and challenges," *Comput. Appl. Eng. Educ.*, 2022.
- [8] L. Tawalbeh, F. Muheidat, M. Tawalbeh, M. Quwaider, and A. A. Abd El-Latif, "Edge enabled IoT system model for secure healthcare," *Measurement (Lond.)*, vol. 191, no. 110792, p. 110792, 2022.
- [9] B. Costa, J. Bachiega Jr, L. R. de Carvalho, and A. P. F. Araujo, "Orchestration in fog computing: A comprehensive survey," *ACM Comput. Surv.*, vol. 55, no. 2, pp. 1–34, 2023.
- [10] A. Volkov, K. Proshutinskiy, A. B. M. Adam, A. A. Ateya, A. Muthanna, and A. Koucheryavy, "SDN load prediction algorithm based on artificial intelligence," in *Communications in Computer and Information Science*, Cham: Springer International Publishing, 2019, pp. 27–40.
- [11] K. Kaur, V. Mangat, and K. Kumar, "A review on Virtualized Infrastructure Managers with management and orchestration features in NFV architecture," *Comput. Netw.*, vol. 217, no. 109281, p. 109281, 2022.
- [12] W. Lv et al., "Microservice deployment in edge computing based on deep Q learning," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 11, pp. 1–1, 2022.