http://dx.doi.org/10.5755/j01.eie.26.1.23235 ELEKTRONIKA IR ELEKTROTECHNIKA, ISSN 1392-1215, VOL. 26, NO. 1, 2020

Performance Analysis of Edge Computing in IoT

Julius Skirelis*, Dalius Navakauskas
Department of Electronic Systems, Vilnius Gediminas Technical University,
Naugarduko St. 41-413, LT-03227 Vilnius, Lithuania
julius.skirelis@vgtu.lt

***Abstract*—Upcoming 5G technology and the demand for near real-time response services raise the need for optimizing current IoT solutions. The aim of this paper is to model and execute the performance analysis of network structures suitable for Edge Computing in IoT. The prior research into different topology and parameter sets have not provided sufficient clarity, on which parameters had a considerable impact on overall system performance; therefore, repetitive simulations were performed to express dispersion of alternating values, as well as determining its confidence intervals. The paper presents Edge Computing service simulation setup on known and newly derived network topologies with different complexity varying network bandwidth and network delay parameters. The experimental investigation has revealed that the IoT configuration network is more sensitive to network topology, while the Internet configuration is more sensitive to network parameters. The discussion on the results received debates possible causes of performance differences in network parameters and device configurations, the comparison to similar state-of-the-art research results has also been presented. Finally, conclusions with recommendations based on the results acquired have been provided.**

***Index Terms*—Internet of things; Edge computing; Computer network reliability.**

I. INTRODUCTION

Upcoming 5G technology and the demand for near real-time response services raise the need for optimizing current IoT solutions. The demand for cloud technologies for big data processing is tremendous; the insights into [1], [2] show that with the realization of Industry 4.0 it would not be possible to process such enormous data capacities. The Edge Computing technology is highly suitable for low-power devices, such as wearables, mobile gadgets or even sensors themselves [3] to adopt computing tasks from common cloud technology. Local or even global internetworking of physical devices (things) introduces new constraints on the IoT paradigm.

The most important undertaking in superior Edge Computing realization is an appropriate job scheduling strategy selection [4] aiming to fulfil timeliness criteria, i.e., to perform a task before its data became obsolete [5]. However, to support this, the proper law of reorganization of a dynamic network architecture must be determined.

Numerous studies investigate the job-scheduling problem by proposing algorithms for the optimal virtual machine (VM) placement across the computer network. However, most of them do not consider the optimization of network architecture [6]. Therefore, in this paper, we describe modelling and performance analysis of different topologies with an idea to propose a dynamic self-organizing network topology in order to increase the performance of Edge Computing.

The paper is structured as follows. Related research is discussed in Section II. Experimental simulation setup, including the preparation of data and parameters, is given in Section III, and the results of state-of-the-art comparability simulations are discussed in Section V. Conclusions of the results are presented and analyzed in Section IV.

II. PREVIOUS RESEARCH

Research on Mobile Edge Computing optimization aims at energy efficiency improvements [7]–[9] and decentralized computation offloading [10]. Our preliminary study of Edge Computing [7] performed with *EdgeNetworkCloudSim* tool did provide overlapping results suggesting the need for executing more detailed and multiple simulations. The structure of the experimental study in this article corresponds to the previous article [7], and the researched topologies are common to the paper published by the creators of simulator; we modified the topologies to emphasize the impact.

A similar study [11] on Edge Computing services for cars on highways used NS-2 software for a video-on-demand service simulation confirming the proposed architecture suitability for LTE mobile systems. The authors provided a comparison and performance analysis in scenarios for 2-edge and 4-edge devices, as well as discussing the terms of the average number of received packets and service time.

A self-organizing framework for Edge Computing task scheduling was proposed in [12]. It employs a continuous Complex Event Processing technique, where each task is decomposed to Direct Acyclic Graphs, while scheduling takes into account network loads and edge device loads. However, intelligent measures were not used in the network topology reorganization there.

A classic Petri net model was used in a healthcare-oriented framework investigation [13], where an improved model, Resource Preservation Net, was proposed. It includes edge network resource handling with the ability to reorganize edge device role using Artificial Intelligence in a

Manuscript received 17 April, 2019; accepted 15 January, 2020.

workflow. Analysis and estimation of the Quality of Experience confirmed that the performance of the Edge Computing application is highly sensitive on task scheduling technique, as well as on network parameters or their brokering methods. On the basis of VM mitigation, server resources and task complexity evaluation, the online and offline edge computing network offloading algorithms were proposed [14], [15]. They include path selection, tier decomposition, Ad Hoc cloud-assisted and partitioning techniques. One of these techniques based on a distributed hardware CNN classifier in satellite research [16] showed up to 99.4% success ratios in learning *QoE* values.

According to *EdgeNetworkCloudSim* creators report [17] and performance evaluations of *NetworkCloudSim* (the base tool) [18], the simulation results lack the scope of application and determination of the parameters used. That is why on the basis of environments defined in previous research [19], [20], we tune parameters and configurations for certain fields of application, namely, Industry 4.0 and Cyber-Physical production systems. Determining the subject for optimization (the impact of parameters on the performance) is a crucial task towards the implementation of Edge Computing-based systems in the fields mentioned before.

III. INVESTIGATION SETUP

The virtual experimental environment based on *EdgeNetworkCloudSim* tool was set up to perform simulations. The two cases of the simulation were arranged: the Internet and the IoT.

A. Experimental Environment

Simulations were performed on *Java* virtual machine on *Ubuntu 16.04* operating system. A *Java* virtual machine ensures isolation, thus guaranteeing that random initiation and simulation execution is operating system independent. The analysis of results was performed with *Matlab* software. The network topology was generated by *BRITE* tool.

B. Controllable Parameters

Network topologies, including delay and network capacity parameters between nodes, are defined by *BRITE* structure. In the simulation software, each node resource for Internet and IoT is defined by the following parameters: million instructions per second (MIPS), the capacity of random-access memory (RAM), and the bandwidth and amount of data that can be stored in a particular node. For the Internet case, M0 device configuration was used for simulation on all nodes, while for IoT devices three types of configurations M1–M3 were applied. Device parameters were selected on the basis of the real hardware used in Industrial Edge Computing studies [21], [22] and Cyber-Physical systems [23], [24] used for studying connected automated vehicles, together with large scaled video analysis platforms; their values are presented in Table I.

TABLE I. EDGE COMPUTING DEVICE CONFIGURATIONS.

Name	MIPS	RAM, MB	Bandwidth, kB/s	Storage, MB
M0	10,000	4,096	5,120	10,240
M1	2,000	512	1,024	1,024
M2	1000	128	1,024	128
M3	100	16	512	64

Device M1 represents *Raspberry PI 3* hardware, M2 is a *LinkIt Smart 7688 Du* IoT device, and M3 is *ESP32* SoC. These values represent the maximum values of the parameters the device can reach. The assignment of configurations to IoT devices was done in random order.

The Edge Computing device configuration parameters were selected to represent typical low-efficiency IoT devices realistically, as well as an average consumer PC.

C. Varied Parameters

For the Internet and IoT comparison purposes, the three sets of network topologies were defined.

Figure 1 illustrates the IoT network topology with 13 edge devices and 15 nodes denoted by A_0. This topology serves as a reference since it has been analyzed by others [17]. In order to reveal the sensitivity of the IoT network structure, the additional 16th node was added (see a brown dotted node in Fig. 1).

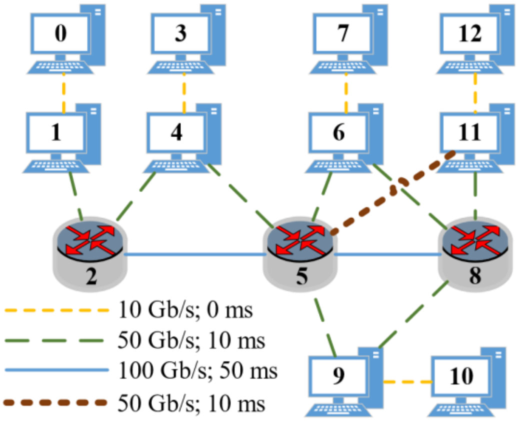

Fig. 1. Pictorial representation of A_0 and A_1 network topologies having 13 nodes with 15 and 16 edges.

Consequently, the 8th node device is offloaded and the topology denoted by A_1 becomes fully connected. Given that numerous topologies can be defined, the purpose of this study is not to discover the best one. That is why only a single alternative topology denoted by B with 22 edges and 27 node links (Fig. 2), was generated. Topology B is 30% more complex than previous topologies (A_0 and A_1).

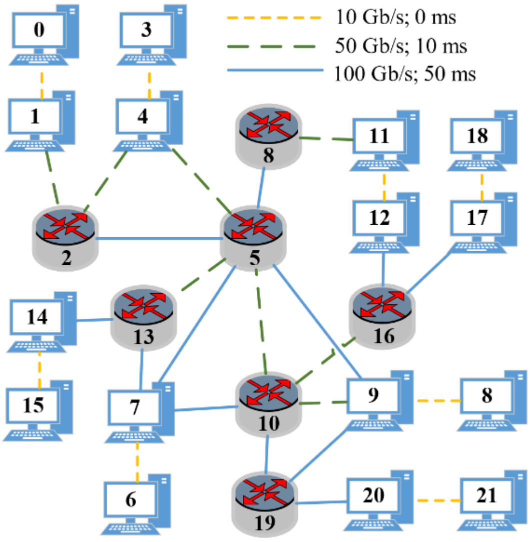

Fig. 2. Pictorial representation of B network topology having 22 nodes with 27 edges.

In all network topologies, device *0* is a packet initiator device exclusively having following parameters: up to 1 Gbps throughput with 5 ms delay for Internet and 54 Mbps throughput with 100 ms delay (IEEE 802.11g) for IoT, respectively. As a network load, data chunks with a size of 1 GB were used.

Three link parameter sets are varied: baseline denoted as P_0 (its values for corresponding links are provided in Fig. 1 and Fig. 2); double bandwidth, $P_{2\mathrm{B}}$ doubling baseline bandwidth value; and double delay, $P_{2\mathrm{D}}$ doubling baseline delay value accordingly.

A dynamic fluctuation of the network bandwidth, a jitter, was imposed between nodes. Random values g_i were generated using Gaussian distribution (μ = 0; σ = 1) and then used as a jitter multiplier in order to simulate the fluctuations

$$B_i(n) = g_i \times B_0(n), \tag{1}$$

where $B_0(n)$ denotes the bandwidth of nth node, g_i is a jitter coefficient following Gaussian distribution at ith experiment; n is a node index, and i is an experiment index. All specified experiments were repeated 100 times. Figure 3 presents an exemplary histogram of used jitter coefficients selected within a reasonable [−0.03, 0.03] range.

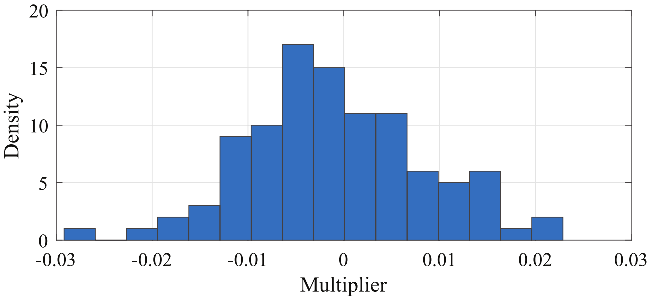

Fig. 3. Exemplary 100 sample histogram of bandwidth jitter coefficients used during experiments.

IV. Investigation Results

After performing a total of 1,800 simulations for each of two network configurations and three different topologies with varying bandwidth and delay parameters (baseline, double bandwidth, and double delay) repeated 100 times, the investigation procedure was performed.

A. *Procedural Steps*

The distribution of individual experiment estimates of packet delivery success ratios was found to be Gaussian. That is why the success of packet delivery events is evaluated by estimating the ratio values:

$$E_i^{\mathrm{S}}(\mathcal{T},\mathcal{P}) = \frac{E_i^{\mathrm{S}}(\mathcal{T},\mathcal{P})}{E_i^{\mathrm{T}}(\mathcal{T},\mathcal{P})} \times 100\%; \tag{2}$$

together with mean and standard deviation of success ratio values as follows:

$$E_{\mathrm{m}}^{\mathrm{S}}(\mathcal{T},\mathcal{P}) = \frac{1}{N}\sum_{i=1}^{N} E_i^{\mathrm{S}}(\mathcal{T},\mathcal{P}), \tag{3}$$

$$E_{\mathrm{d}}^{\mathrm{S}}(\mathcal{T},\mathcal{P}) = \operatorname*{std}_{i\in[1,N]} E_i^{\mathrm{S}}(\mathcal{T},\mathcal{P}), \tag{4}$$

where $E_i^{\mathrm{S}}(\mathcal{T},\mathcal{P})$ and $E_i^{\mathrm{T}}(\mathcal{T},\mathcal{P})$ are the numbers of successful packet delivery and total events during ith experiment given a specific network configuration $\mathcal{T} \in \{A_0, A_1, B\}$ and specific set of parameters $\mathcal{P} \in \{P_0, P_{2\mathrm{D}}, P_{2\mathrm{B}}\}$, E_i^{S} is an estimate of packet delivery success ratio during ith experiment, $E_{\mathrm{m}}^{\mathrm{S}}$ and $E_{\mathrm{d}}^{\mathrm{S}}$ are mean and dispersion values of packet delivery success ratio estimates, i is an experiment index, and N is a total number of experiments, N = 100.

The distribution of individual experiment estimates of packet delivery failure ratios was found to be exponential. As a result, the failure of packet delivery events is evaluated with a 95% confidence interval by estimating the ratio values

$$E_i^{\mathrm{F}}(\mathcal{T},\mathcal{P}) = \frac{E_i^{\mathrm{F}}(\mathcal{T},\mathcal{P})}{E_i^{\mathrm{T}}(\mathcal{T},\mathcal{P})} \times 100\%, \tag{5}$$

and mean together with variance values of failure ratio as follows:

$$E_{\mathrm{m}}^{\mathrm{F}}(\mathcal{T},\mathcal{P}) = \frac{1}{N}\sum_{i=1}^{N} E_i^{\mathrm{F}}(\mathcal{T},\mathcal{P}), \tag{6}$$

$$E_{\mathrm{v}}^{\mathrm{F}}(\mathcal{T},\mathcal{P}) = \operatorname*{var}_{i\in[1,N]} E_i^{\mathrm{F}}(\mathcal{T},\mathcal{P}), \tag{7}$$

where E_i^{F} is the number of packet delivery failures, while E_i^{F} is an estimate of packet delivery failure ratio during the ith experiment, $E_{\mathrm{m}}^{\mathrm{F}}$ and $E_{\mathrm{v}}^{\mathrm{F}}$ are mean and variance values of packet delivery failure ratio estimates.

The status difference for success (8) and failure (9) cases between IoT and Internet network configurations is expressed as an algebraic difference between mean ratio values:

$$\Delta E_{\mathrm{m}}^{\mathrm{S}}(\mathcal{T},\mathcal{P}) = E_{\mathrm{m}}^{\mathrm{S}^{\mathrm{IoT}}}(\mathcal{T},\mathcal{P}) - E_{\mathrm{m}}^{\mathrm{S}^{\mathrm{Int}}}(\mathcal{T},\mathcal{P}), \tag{8}$$

$$\Delta E_{\mathrm{m}}^{\mathrm{F}}(\mathcal{T},\mathcal{P}) = E_{\mathrm{m}}^{\mathrm{F}^{\mathrm{IoT}}}(\mathcal{T},\mathcal{P}) - E_{\mathrm{m}}^{\mathrm{F}^{\mathrm{Int}}}(\mathcal{T},\mathcal{P}). \tag{9}$$

By analogy, service duration statistics were analyzed, the inverse exponential distribution was obtained, corresponding estimation distribution was applied, and statistical measures of mean and variance values were calculated as follows:

$$D_{\mathrm{m}}(\mathcal{T},\mathcal{P}) = \frac{1}{N}\sum_{i=1}^{N} D_i(\mathcal{T},\mathcal{P}), \tag{10}$$

$$D_{\mathrm{v}}(\mathcal{T},\mathcal{P}) = \operatorname*{var}_{i\in[1,N]} D_i(\mathcal{T},\mathcal{P}), \tag{11}$$

where D_i is the duration of service execution during the ith experiment given specific network configuration $\mathcal{T}$ and specific set of parameters $\mathcal{P}$, D_{m}, and D_{v} are the mean and variance values of service execution durations.

B. *The Analysis of Results*

Packet delivery states for three different topologies within the two different network configurations, Internet and IoT, are presented in Fig. 4.

The mean success and failure ratios together with their dispersions are compared to different parameter sets: base, double bandwidth, and double delay. The results show that:

– Double bandwidth or delay for the Internet configuration results in narrowing a distribution of successful packets, while the mean ratio value changes slightly;

– The failure ratio on the Internet configuration is more sensitive to the doubled bandwidth or delay and results in an increase of the packet failure ratio;

– Dispersion value is dependent on the mean ratio value and shows the lowest distribution for base parameters, whereas the double delay has the highest impact on the dispersion increase.

Figure 5 presents the integral difference between IoT and Internet network configurations in terms of mean success and failure ratios of packet delivery. It is seen that an increase in success rate decreases the failure rate in 5 of the 9 cases; these events are unrelated and are not the opposite measures.

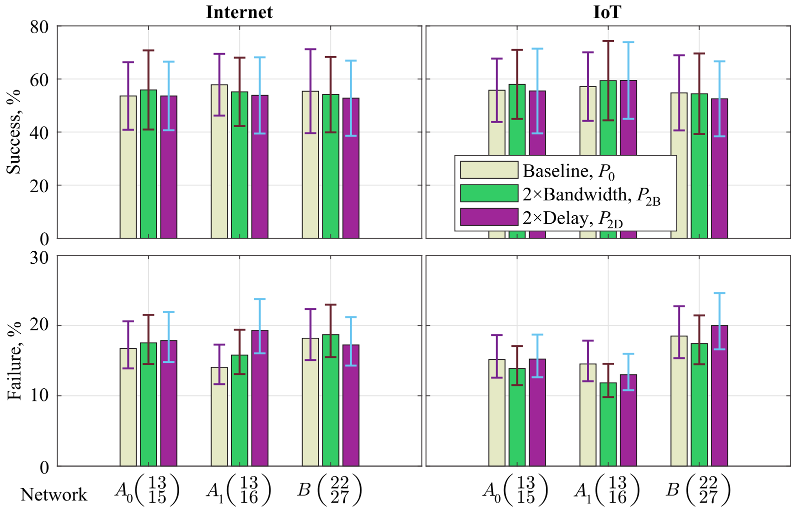

Fig. 4. The results of success and failure ratios represented as mean and standard deviation values for 100 simulation repetitions over different configurations and varying parameter sets of networks.

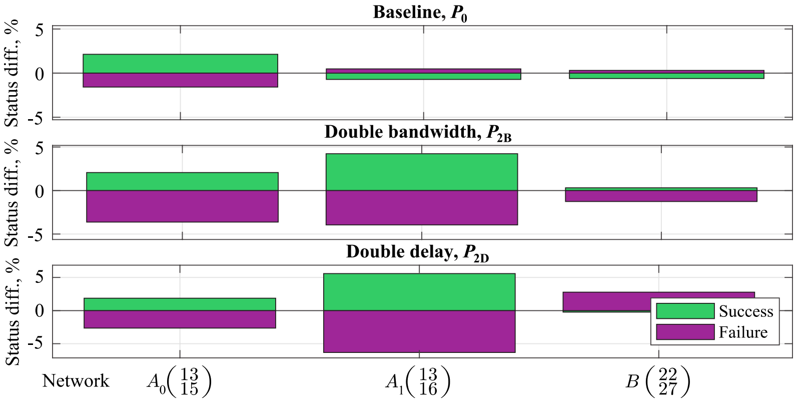

Fig. 5. Mean packet delivery status (success, failure) difference results over 100 simulation repetitions emphasizing network topology and influence of parameter sets on different networks.

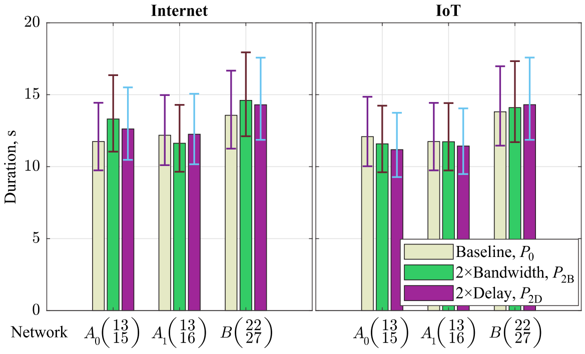

Fig. 6. Mean service duration and variance results of inverse exponential distribution estimation per different topology and parameter sets over 100 simulation repetitions.

Mean service duration and variance values presented in Fig. 6 further demonstrate that, in most cases, worsening network conditions for IoT configurations results in the decreased mean value; however, the distribution of extremums is increasing and in mean value it is much higher than in Internet configuration. It is also evident that the IoT configuration is more sensitive to network topology than to parameter fluctuation. In the majority of cases, the Internet configuration duration is dependent more on the bandwidth available than on the increased delay.

Figure 7 reveals the service duration differences between IoT and Internet network configurations and resulting insights:

– 22-edge and 27-node topology is more sustainable to fluctuations of network parameters across all simulated cases;

– Non-fully scalable Edge Computing is more resistant to the fluctuation of network parameters and results in better mean packet delivery duration than Internet configuration;

– The fluctuation of packet delivery duration difference in a base configuration is negligible; it confirms that under ideal conditions, there exists an insignificant difference between IoT and Internet topologies.

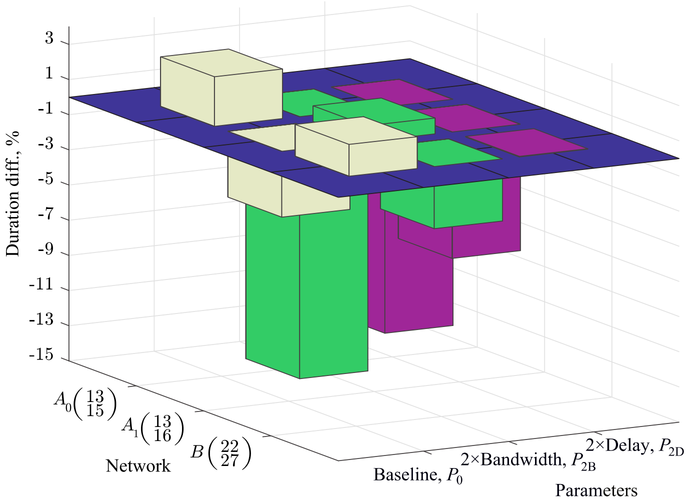

Fig. 7. Service duration difference results between the Internet and IoT configuration, including duration difference on different topologies and different parameter sets.

V. DISCUSSION

The first investigation (packet delivery status) has supported the idea that device configuration has a little impact on successfully delivered packets [18], its mean ratio value to total events occurred in the routing activity fluctuated near 50% for both Internet and IoT configurations. To reveal the differences, the distribution value (Fig. 4 error bars) has been considered. It is evident that varied network parameters start to reflect in successful packets ratio value, thus doubling the delay; the probability of successful packet delivery increases to 65% and even more. The failure ratio statistics confirm the principle: the more bandwidth and fewer delays, the fewer failing packets [23]. Our contribution has not submitted any new insights based on distribution differences in terms of failing packets ratio. Such a behavior summarized in a differential view (Fig. 5) leads to a recommendation on using fully scalable network topologies in the Edge Computing systems for increasing the success packet count and decreasing the failing packet count. It should be noted here, that apart from success and failure, there exist additional packet delivery states, which have not been taken into account, and which, in our opinion, do not have an impact on the system performance.

The results of the second investigation (service duration analysis) partially coincide with the results obtained in the related research [17] stating that the double delay parameter increases the total service time in the same topology. However, we have obtained slightly different results in terms of double bandwidth parameter; the related research experiments have shown that it has no effect on the service time, but that is true only when investigating the single device configuration. That is why the two device categories, Internet and IoT, have been established. It has been revealed that on highly resourced Internet configuration exploits a fully scalable network contributes to decreasing the service time, while in IoT case, as we assume, hardware limits occur, and the service time decrease does not happen in topology A_1 with double bandwidth. This raises a hypothesis that network stack is capable of adapting itself to an exhausted network and/or device resource limit conditions producing maximum performance with priority on hardware resources. For this reason, with Fig. 7 we wanted to show how the node device hardware affects on the same network topology under exact same conditions. Surprisingly, the service duration value of baseline conditions varied in a ±5% margin, and the best network utilization scenario is seen over small network topologies confirming the recommendation from the related research [23] on reducing and segmenting the schema (topology in our case) in order to increase the system performance.

Additionally, we would like to note that the experiment has been performed in a virtual environment with real network and device parameters, together with related studies that prove conformity with the simulator used [17], [18]. Close-to-real conditions have been assured to be simulated, and the results acquired have been very close to reality. In order to confirm the results of these experiments on real hardware, further research is needed.

VI. CONCLUSIONS

Simulation and performance analysis of the modelled Edge Computing topology has been performed. A total of 100 repetitive simulations over the controlled bandwidth jitter have been managed.

The results received using *EdgeNetworkCloudSim* package confirm that:

1. Differentiating delay value has the greatest impact on a packet routing status (success, failure);
2. Fully scalable network topology is less susceptible to network bottlenecks;
3. Increasing bandwidth and delay values of links result in an increased failure rate, while the distribution of mean values within successful routes is decreasing.

The above-mentioned conclusions and practical insights into the performed experiment commend model Edge

Computing systems with the lowest link delay value possible since delay has a direct impact on overall system stability.

Developers of embedded (Industry 4.0) systems should focus on network quality and reachability on edge devices mainly, i.e., fully scalable or partially reroutable paths in a network arrangement rather than on hardware resources. The aforementioned statement is valid when dealing with low load (≤ 30 nodes) services.

In jittery conditions (in terms of bandwidth), the solution for increasing the performance of the system is to scale down the network and/or split the whole topology into segments, i.e., subnets in order to maintain the principle of locality.

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

[1] S. Ahmad *et al.*, “Towards the design of a formal verification and evaluation tool of real-time tasks scheduling of IoT applications”, *Sustainability*, vol. 11, no. 1, p. 204, Jan. 2019. DOI: 10.3390/su11010204.

[2] W. Na, Y. Lee, N.-N. Dao, D. N. Vu, A. Masood, and S. Cho, “Directional link scheduling for real-time data processing in smart manufacturing system”, *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3661–3671, Oct. 2018. DOI: 10.1109/JIOT.2018.2865756.

[3] M. Ali, N. Riaz, M. I. Ashraf, S. Qaisar, and M. Naeem, “Joint cloudlet selection and latency minimization in fog networks”, *IEEE Trans. Ind. Informatics*, vol. 14, no. 9, pp. 4055–4063, Sep. 2018. DOI: 10.1109/TII.2018.2829751.

[4] I.-D. Filip, F. Pop, C. Serbanescu, and C. Choi, “Microservices scheduling model over heterogeneous cloud-edge environments as support for IoT applications”, *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2672–2681, Aug. 2018. DOI: 10.1109/JIOT.2018.2792940.

[5] R. Ghosh and Y. Simmhan, “Distributed scheduling of event analytics across edge and cloud”, *ACM Trans. Cyber-Physical Syst.*, vol. 2, no. 4, pp. 1–28, Jul. 2018. DOI: 10.1145/3140256.

[6] A. Kiani and N. Ansari, “Edge computing aware NOMA for 5G networks”, *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1299–1306, Apr. 2018. DOI: 10.1109/JIOT.2018.2796542.

[7] J. Skirelis and D. Navakauskas, “Edge computing in IoT: Preliminary results on modeling and performance analysis”, in *Proc. of 2017 5th IEEE Work. Adv. Information, Electron. Electr. Eng.*, 2017, pp. 1–4. DOI: 10.1109/AIEEE.2017.8270555.

[8] F. Wang, J. Xu, X. Wang, and S. Cui, “Joint offloading and computing optimization in wireless powered mobile-edge computing systems”, *IEEE Trans. Wirel. Commun.*, vol. 17, no. 3, pp. 1784–1797, 2018. DOI: 10.1109/TWC.2017.2785305.

[9] P. Liu, G. Xu, K. Yang, K. Wang, and X. Meng, “Jointly optimized energy-minimal resource allocation in cache-enhanced mobile edge computing systems”, *IEEE Access*, vol. 7, pp. 3336–3347, 2019. DOI: 10.1109/ACCESS.2018.2889815.

[10] S. Josilo and G. Dan, “Selfish decentralized computation offloading for mobile cloud computing in dense wireless networks”, *IEEE Trans. Mob. Comput.*, vol. 18, no. 1, pp. 207–220, Jan. 2019. DOI: 10.1109/TMC.2018.2829874.

[11] S. Lee, J. Lee, H. Cho, “A study of mobile edge computing system architecture for connected car media services on highway”, *KSII Trans. Internet Inf. Syst.*, vol. 12, no. 12, Dec. 2018. DOI: 10.3837/tiis.2018.12.004.

[12] S. Choochotkaew, H. Yamaguchi, and T. Higashino, “A self-organized task distribution framework for module-based event stream processing”, *IEEE Access*, vol. 7, pp. 6493–6509, 2019. DOI: 10.1109/ACCESS.2018.2890005.

[13] S. Oueida *et al.*, “An edge computing based smart healthcare framework for resource management”, *Sensors*, vol. 18, no. 12, p. 4307, Dec. 2018. DOI: 10.3390/s18124307.

[14] J. Wang, J. Pan, F. Esposito, P. Calyam, Z. Yang, and P. Mohapatra, “Edge cloud offloading algorithms: Issues, methods, and perspectives”, *ACM Comput. Surv.*, vol. 52, no. 1, pp. 1–23, Feb. 2018. DOI: 10.1145/3284387.

[15] R. K. Verma, K. K. Pattanaik, S. Bharti, and D. Saxena, “In-network context inference in IoT sensory environment for efficient network resource utilization”, *J. Netw. Comput. Appl.*, vol. 130, pp. 89–103, Mar. 2019. DOI: 10.1016/j.jnca.2019.01.013.

[16] J. Kang and D.-S. Eom, “Offloading and transmission strategies for IoT edge devices and networks”, *Sensors*, vol. 19, no. 4, p. 835, Feb. 2019. DOI: 10.3390/s19040835.

[17] M. Seufert, B. K. Kwam, F. Wamser, and P. Tran-Gia, “EdgeNetworkCloudsim: Placement of service chains in edge clouds using NetworkCloudsim”, in *Proc. of 2017 IEEE Conference on Network Softwarization: Softwarization Sustaining a Hyper-Connected World: en Route to 5G, NetSoft 2017*, 2017. DOI: 10.1109/NETSOFT.2017.8004247.

[18] S. K. Garg and R. Buyya, “NetworkCloudSim: Modelling parallel applications in cloud simulations”, in *Proc. of 2011 4th IEEE International Conference on Utility and Cloud Computing, UCC 2011*, 2011, pp. 105–113. DOI: 10.1109/UCC.2011.24.

[19] R. Beregi, G. Pedone, and I. Mezgár, “A novel fluid architecture for cyber-physical production systems”, *International Journal of Computer Integrated Manufacturing*, vol. 32, no. 4–5, pp. 340–351, May 2019. DOI: 10.1080/0951192X.2019.1571239.

[20] T. Tomiyama and F. Moyen, “Resilient architecture for cyber-physical production systems”, *CIRP Annals*, vol. 67, no. 1, pp. 161–164, 2018. DOI: 10.1016/j.cirp.2018.04.021.

[21] P. J. Basford *et al.*, “Performance analysis of single board computer clusters”, *Future Generation Computer Systems*, vol. 102, pp. 278–291, Jan. 2020. DOI: 10.1016/j.future.2019.07.040.

[22] Y. Yu, R. Chen, H. Li, Y. Li, and A. Tian, “Toward data security in edge intelligent IIoT”, *IEEE Network*, vol. 33, no. 5, pp. 20–26, Sep. 2019. DOI: 10.1109/MNET.001.1800507.

[23] J. Tang *et al.*, “π-Hub: Large-scale video learning, storage, and retrieval on heterogeneous hardware platforms”, *Future Generation Computer Systems*, vol. 102, pp. 514–523, Jan. 2020. DOI: 10.1016/j.future.2019.08.006.

[24] Y. Feng, B. Hu, H. Hao, Y. Gao, Z. Li, and J. Tan, “Design of distributed cyber-physical systems for connected and automated vehicles with implementing methodologies”, *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 4200–4211, Sep. 2018. DOI: 10.1109/TII.2018.2805910.