Network Load Balancing for Edge-Cloud Continuum Ecosystems

Andrzej Paszkiewicz[1(✉)], Marek Bolanowski[2], Cezary Ćwikła[1], Maria Ganzha[2], Marcin Paprzycki[2,3], Carlos E. Palau[4], and Ignacio Lacalle Úbeda[4]

[1] Rzeszow University of Technology, al. Powstańców Warszawy 12, 35-595 Rzeszów, Poland
andrzejp@prz.edu.pl
[2] Systems Research Institute Polish Academy of Sciences, Warszawa, Poland
marekb@prz.edu.pl
[3] Galgotias University, Greater Noida, India
[4] Universitat Politécnica de Valencia, Valencia, Spain

AQ1

Abstract. This contribution concerns load balancing, based on mechanisms from complex systems theory dedicated to IoT solutions within the Edge-Cloud continuum. The basis of considered mechanisms is the betweenness analysis applied to distributed nodes in a wireless IoT network. A high value of this parameter can indicate the key role of a given node, which is often reflected in its high load. In addition, both the distance and the error rate for connections between nodes are considered. The proposed solution aims at providing path redundancy in the wireless network and enable efficient distribution of the network traffic load.

Keywords: Network load balancing · Complex systems · Edge-cloud continuum · IoT ecosystems

1 Introduction

With the development of a digital society, the number of devices connected to the network infrastructure is rapidly increasing [1], as well as services that make use of the information coming from those devices. Application areas cover, among others, industry, medicine, homes, transportation and urban infrastructures [2–6]. At the same time, the ongoing miniaturization makes it possible to deploy highly heterogeneous Internet of Things (IoT) ecosystems. This heterogeneity results in the need to depart the initial vision of cloud-centered IoT. Instead, the Edge-Cloud Computing (ECC) has been proposed [7]. Here, data can be processed in any node of the ecosystem to realize a user-defined workflow and deliver services. To realize this approach, it is necessary to establish a reliable interconnection infrastructure, ensuring consistency and redundancy of data transmission paths. Moreover, load balancing plays an important role as well. These mechanisms are a set towards ensuring the efficiency of the ecosystems' networks [8, 9].

S. Mekhilef et al. (Eds.): ICEEE 2022, LNEE 894, pp. 1–14, 2022.
https://doi.org/10.1007/978-981-19-1677-9_56

Problems related to load balancing in computer networks has been studied for many years [10, 11]. The primary goal is to evenly distribute traffic across multiple paths/nodes and even servers like in Content Distribution Networks and Cache server infrastructures [12]. Mechanisms of this type often guarantee infrastructure redundancy [13]. Overall, optimization of network traffic, and efficient utilization of network resources, achieved by load balancing positively affects system scalability and minimizes delays and packet losses. There are two approaches to solve load balancing problem: static and dynamic. Static approaches are applicable mainly when all necessary information and parameters are known and available beforehand. In dynamic mechanisms, the necessary data is provided during network operation. Traditionally, load balancing has been based on vendor-lock, device (i.e., switch) programmable mechanisms, that entail hardware burdens and limit the scope of dynamic balancing. Lately, emerging technologies like Software Defined Networks (SDN) [14] and Networks Functions Virtualization (NFV) facilitate these operations, relying on the distribution of network traffic based on open, modifiable software rules [15].

Nowadays, three main areas of application of load balancing mechanisms can be identified: wired networks, server infrastructure and ECC infrastructures. For wired networks, load balancing mechanisms have been used primarily at Layer 2 and Layer 3 of the ISO/OSI model. These include Link Aggregation Control Protocol, Spanning Tree Protocol, or in Equal-Cost Multi-Path routing. Moreover, new approaches, including load balancing in SDNs [16, 17], have been proposed. Server Load Balancing [18] has many implementations, involving e.g., network switches, dedicated appliances, DNS re-direction or virtualization platforms [19]. Moreover, cloud-based and NFV load balancing are also relevant [20]. Here, note that solutions dedicated to server systems often find application in Clouds, and vice versa. On the other hand, ECC ecosystems bring new challenges. Here, as of today, typical solutions are based on the assumption of existence of a network gateway, which connects the wireless IoT network with the wired network. Then, load balancing system provides redundant connections to this gateway. On the other hand, the development of wireless IoT networks, especially for autonomous sensor systems, should also ensure redundancy of connections between elements of the sensor-actuator infrastructure. This observation provides the context for the current contribution.

2 Related Work

The diverse purpose of networks within ECC ecosystems, causes problems in defining common standards for specific functionalities. An example of such functionality is load balancing. Here, the created mechanisms must meet different requirements related to e.g. delay/RTT (Round Trip Time) minimization, path redundancy, packet loss tolerance or battery use control. Hence, various parameters are used to define operation and evaluation of load balancing mechanisms.

One such mechanism is *network lifetime*. It indicates the time the network operates from the start, until the energy of the first node, a given percentage of nodes, and the last node is exhausted [21]. This is relevant for networks performing measurement tasks in a specific time interval. Another parameter considered in load balancing is *fault* *tolerance*, as related to the recovery of network infrastructure after a link, or node, failure. It is crucial for networks required to operate continuously [22]. Another important parameter is *scalability*. It can be seen as the ability to adapt to changes taking place in the environment [23], the capacity to accept new users over the same infrastructure [24] or as an indicator of the system's ability to load-balance in a network with a limited number of servers [25], considering that scalability in IoT environments is directly linked with density and k-connectivity. Load balancing can be also seen from the perspective of *reliability*, i.e., probability of a successful service operation [26]. Another way to look at this issue is to analyze the *packet loss ratio* [27] or the *delivered packet ratio* [28]. Here, the impact of the quality and level of load is correlated with the increase or decrease of these parameters. On the other hand, latency is also analyzed from the perspective of load balancing mechanisms [29]. In this case, it depends on many factors such as number of hops, node density, k-connectivity or data rates [30]. Quality of load balancing can be also evaluated from the perspective of *energy consumption* with a direct impact in routing strategies. For example, influence of the load of cluster structure elements on the energy consumption level was studied in [31]. An interesting solution is the IPv6 routing protocol for Low-Power and Lossy Networks (RPL) [32]. This work is based on a tree-like structure, where each network node is assigned a rank that changes depending on its position relative to the main node. In the proposed protocol, the concept of an instance is used, which refers to multiple directed acyclic graphs (DAGs) using the same routing rules and mechanisms. However, while allowing creation of redundant routes RPL encounters problems with the emergence of high network load. To eliminate the problems of classic RPL protocol, Context-Aware approach has been proposed [33]. The main idea of this approach is to reduce packet loss and take care of network lifetime by considering the surroundings of a node. Further enhancements of the RPL-based approach were related to the objective function. In [34], a combination of a primary metric, exemplified by the number of nodes represented by the number of hops or the expected transmission count, with a secondary metric based on, e.g., the number of sons for a given parent node, is used. When computing node rankings, and selecting a candidate parent, the objective function uses the primary metric, while the secondary metric is used to reject unsuitable parents.

The IoT environment is designed to integrate multiple objects understood as devices and computing elements along the ECC. One example of such an environment is systems dedicated to e-health solutions. The load balancing algorithm for efficient and reliable communication within e-health environment, proposed in [35], is based on categorization of packet transmission routes in terms of loss and error rate, mobility rate and load factor. This mechanism improves the flow control mechanism and combines some parameters used in the TCP protocol. Another load balancing mechanism applicable to ECC ecosystems is adaptive routing for heterogeneous infrastructure [36]. It aims at maximizing throughput by sharing the load among different gateways in the network. It accomplishes this by limiting the amount of data that gateways can send among themselves, and limits the amount of data received by a particular gateway. The proposed approach selects the next hop dynamically, based on the load factor index of the neighboring gateways.

Based on literature analysis it can be assumed that most load balancing mechanisms, applicable to ECC ecosystems, assume availability of a network gateway(s) with which local units communicate. Interestingly, similar approach was proposed to move vehicles and passengers [37, 38]. Unfortunately, there are no studies addressing directly load balancing between distributed nodes within the Edge-Cloud continuum. Therefore, the proposed solution in this paper addresses these needs.

3 Proposed Solution

It is relatively easy to notice that networks that provide foundation of ECC ecosystems fit the theory of complex systems [39, 40], and that the processes in them are dynamic in nature. Hence, a measure based on *betweenness* [41] can be used. Here, nodes with high betweenness can be considered as key intermediate points (hubs) that play an active role in facilitating communication. Thus, betweenness determines how often a node appears on the shortest path between two nodes and has been defined as:

$$b(v) = \sum_{s \neq v \neq t} \frac{\delta_{st}(v)}{\delta_{st}}, \tag{1}$$

where $\delta_{st}(v)$ is the number of shortest paths from s to t that pass through node v, and δ_{st} is the total number of shortest paths from s to t. High betweenness values mean that a given vertex is located on a significant part of the shortest paths connecting pairs of nodes. Assuming that networks in ECC ecosystems cannot adopt exclusively regular topological structures, values of b can vary significantly between nodes. Given that the betweenness increases with the number of vertices in a network, it should be normalized. Hence, an *ordering parameter* was introduced:

$$n = \frac{(N-1)(N-2)}{2}, \tag{2}$$

where N is the number of nodes. Then, the normalized betweenness becomes:

$$b'(v) = nb(v) = \frac{2b(v)}{(N-1)(N-2)}. \tag{3}$$

At this point, it becomes possible to include the betweenness as one of the criteria for selecting communication route(s) between pairs of vertices. To do so, we define the path cost, between a pair of nodes s and t as:

$$c_{st} = \sum_{v \neq s \neq t} b'(v), \tag{4}$$

where node v belongs to path between a pair of nodes s and t. Obviously, there can be many different paths between any pair of nodes s and t. Therefore, we perform a cost-based ordering of the set of alternative routes in the form $P_{st} = \left\{p_1\left(c^i_{st}\right) \leq p_2\left(c^j_{st}\right) \leq \ldots \leq p_m\left(c^k_{st}\right)\right\}$, where $i, j, k \in C_+$, and $p_i\left(c^k_{st}\right)$ denotes i-th path between s and t with a given cost. This allows selection of several routes, starting from p_1.

The presented approach considers the potential number of communication paths passing through each intermediate node. However, for ECC ecosystems, the distance between individual nodes also plays an important role. Therefore, a parameter in the form of distance between individual nodes is considered in the next approach. Therefore, the proposed path cost calculation method considers the distance between nodes, expressed as:

$$a_{wv} = \frac{\sigma D_{vw}}{D_{MAX}}, \tag{5}$$

where D_{vw} is the current distance between two nodes v and w, D_{MAX} is the maximum range of nodes, while $\sigma \in \langle 0; 1\rangle$ is a weighting factor, introduced to account for the energy cost associated with increasing the distance between nodes.

An additional considered parameter covers error rate testing. The occurrence of errors on wireless links may often force a communication disruption requiring the retransmission of data and consequently lead to delays and even network congestion. Therefore, the consideration of this parameter for ECC ecosystem seems to be reasonable. For this purpose, an error threshold value for a given link has been defined, if it is exceeded, the path containing it is not considered in the load balancing mechanism:

$$ER_{vw}(\Delta t) > ER_{TH}, \tag{6}$$

where $ER_{vw}(\Delta t)$ denotes the error rate in a given time interval, for traffic between nodes v and w, and ER_{TH} is the threshold value. Therefore, when condition (6) is satisfied, a path containing an edge e between nodes is removed from the set of available routes between v and w: $P \backslash p_n \in \{e_{vw}\}$.

4 Results of Performed Simulations

The approach, introduced in Sect. 3, was simulated using Matlab (version R2021b). A sample network topology, consisting of 20 nodes, randomly distributed in a 400 × 400 m^2, was used (see, Fig. 1). The maximum range of each node is 200 m. Noted that the explored values are "general" and are not limited to, or dependent on, any specific technology. Communication technologies, their range and capacity are constantly evolving. However, the proposed mechanisms are assumed to be universal.

For each node, the betweenness measure is calculated according to Eq. (1), followed by normalization based on Eq. (2). Next, a matrix containing all possible routes, between pertinent pairs of nodes is created. This matrix is sorted in ascending order, according to the path cost defined in formula (4). Then, the first three paths are selected. If the path cost value is the same for different routes, routes with fewer nodes are selected first. These are paths where the connections do not overlap.

During the simulation, an error rate value is randomized for each connection. It is assumed that the error rate between 0 and 5% will not cause communication problems (some works even place this number at 10% [42]). In reality, in wireless networks, packet

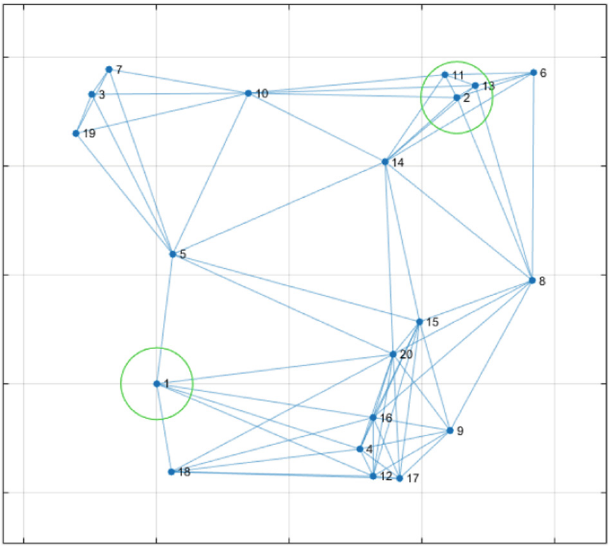

Fig. 1. Network structure analyzed during simulation.

loss due to errors on the link may or may not cause the need for data retransmission. This depends on the type of communication. For only one mechanism simulation, random errors on the link determined the removal of the active load balancing path.

The results for four different load balancing mechanisms are presented, ranging from a solution based only on the betweenness, to a mechanism that additionally consider errors on communication link(s).

4.1 Operation of the Simplest Mechanism (Mechanism 1)

As noted, out of all possible routes, three paths, with the lowest cost, are selected. Here, the betweenness measure does not consider the edge weights, which are the actual distances between the nodes. Three best routes, between nodes 1 and 2, are shown in Table 1, with their costs in the third column, and the number of nodes in the route in the fourth column. Next, Fig. 2 visualizes routes that were selected to forward the packets. Analogous results have been obtained for the remaining connections.

Table 1. Mechanism 1: the best three routes, presented as consecutively visited nodes.

Lp	Node numbers on the path	Path cost	Number of intermediate nodes
1	[1, 18, 17, 9, 8, 2]	0.1950	6
2	[1, 4, 15, 14, 2]	0.3166	5
3	[1, 5, 10, 2]	0.3747	4

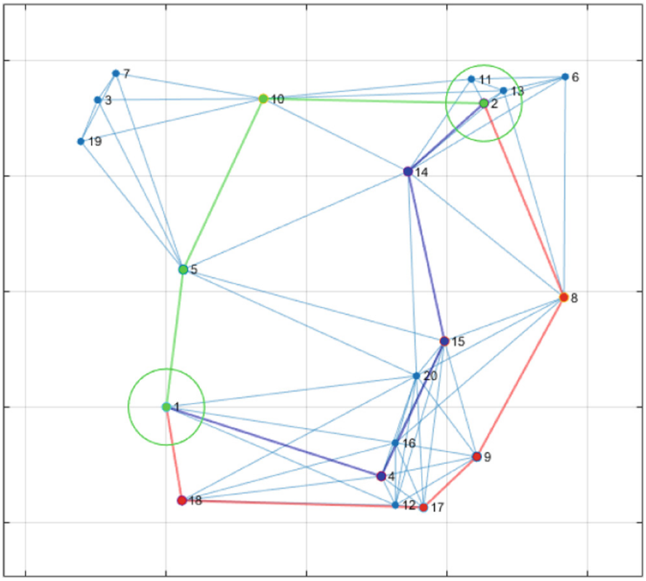

Fig. 2. Visualization of 3 routes, resulting from the simplest mechanism.

In the results, there is a noticeable tendency to choose "edge nodes", as they are rarely part of the shortest paths between any nodes. The advantage of this solution is that it does not overload the "central nodes", which are already heavily loaded. On the other hand, use of paths with a larger number of hops may occur. It may also result in an increase of the load of extreme nodes. The basic mechanism also does not include active response to transmission errors. Here, data retransmission is required, assuming that transmission errors occur only for a short period of time.

4.2 Extending Basic Mechanism with Node Distance (Mechanism 2)

The second mechanism considers the standard distance connecting each node and betweenness values for these nodes when calculating the cost of the route. This considers both the cost associated with the need to provide connectivity between nodes and the load on nodes resulting from the number of potential transmission paths through them. As previously, the best three routes, between nodes 1 and 2, are shown in Table 2, while Fig. 3 visualizes the routes chosen by the mechanism to transmit the packets.

A noticeable change from the first mechanism is the reduction in the number of hops, while avoiding the central nodes. These nodes are usually heavily loaded in the case of classic mechanisms, using only the distance, or the hop metric. After multiple tests, it was noted that, very often, the cost of multiple routes, calculated on the basis of distance and betweenness, is the same. Thus, it can be stipulated that mechanisms, based on the Equal Cost Multi Path principle, can automatically balance the load across different paths without the need for network administrators to manually modify the parameters.

Moreover, in the presented example, the load is better distributed than after applying the basic mechanism. However, the modified mechanism also does not include an active

Table 2. Mechanism 2: the best three routes, presented as consecutively visited nodes.

Lp	Node numbers on the path	Path cost	Number of intermediate nodes
1	[1, 16, 8, 2]	0.0994	4
2	[1, 18, 17, 15, 14.2]	0.3099	6
3	[1, 5, 10, 2]	0.3392	4

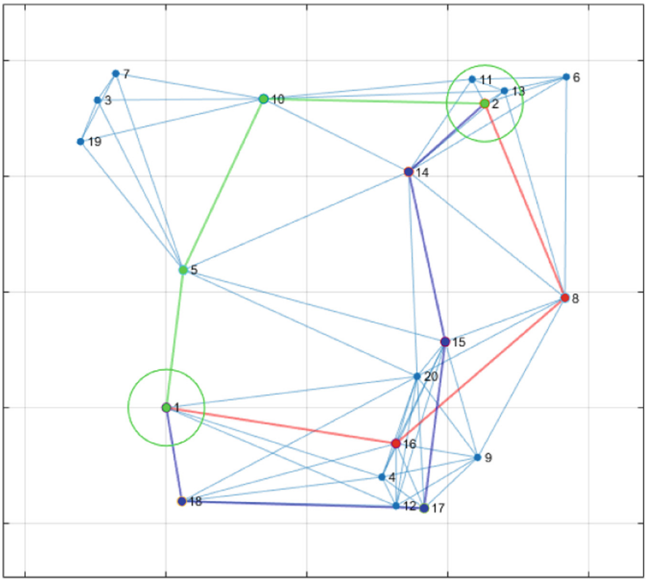

Fig. 3. Visualization of 3 routes selected by the mechanism with node distance.

response to transmission errors. Hence, data retransmission is also required with the assumption that transmission errors only occur for a short period of time.

4.3 Mechanism Which Considers Actual Distance (Mechanism 3)

Again, among possible routes, three paths are selected that have the lowest cost resulting from the betweenness measure. Although, when calculating the cost of the route the actual distance between nodes is also considered, using formula (5).

As previously the best three routes between nodes 1 and 2 are shown in Table 3, while Fig. 4 visualizes the routes that the mechanism selected to forward packets.

Table 3. Mechanism 3: the best three routes, presented as sequentially visited nodes.

Lp	Node numbers on the path	Path cost	Number of intermediate nodes
1	[1, 18, 17, 9, 8, 6, 11, 2]	0.1063	8
2	[1, 4, 16, 15, 14, 2]	0.1776	8
3	[1, 5, 7, 10, 2]	0.1969	5

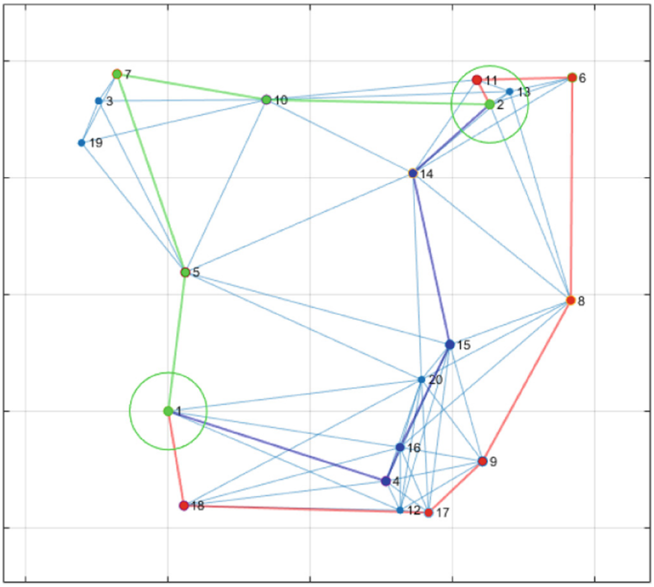

Fig. 4. Visualization of 3 selected routes by the third mechanism.

Mechanism three puts much more emphasis on longer, but more frequent connections, and prefers edge nodes that have very low betweenness values, as can be seen in Fig. 4. This can cause excessive load on intermediate nodes, which will be used more often in communication. The advantage of this approach is in energy savings, as nodes do not have to increase their energy consumption to increase their range, when wanting to send a packet through more distant nodes, and to bypass the central nodes, which are more likely to be significantly overloaded than the edge nodes (in the case of classical mechanisms using only distance or hop metrics). Such an approach is of great importance for sensor networks, where the energy consumption, resulting from increasing the transmitter power may consequently determine (reduce) the lifetime of the network as well as have huge impact in terms of deployment and maintenance cost, environmental degradation, among others.

4.4 Operation of the Mechanism Considering Error Rate (Mechanism 4)

The fourth mechanism considers the distance, when calculating betweenness, as the second mechanism does, and adds an error rate on the links, by examining the error rate for each link. When selecting the best route, if the error rate on an edge exceeds 5%, such edge is temporarily removed and another, best route is obtained. Similarly, the second and third best routes are identified. The top three routes between nodes 1 and 2 are shown in Table 4, while Fig. 5 visualizes the selected routes.

Table 4. Mechanism 4: the best three routes presented as consecutively visited nodes.

Lp	Node numbers on the path	Path cost	Number of intermediate nodes
1	[1, 16, 8, 2]	0.0994	4
2	[1, 18, 17, 15, 14, 2]	0.3099	6
3	[1, 5, 10, 2]	0.6257	5

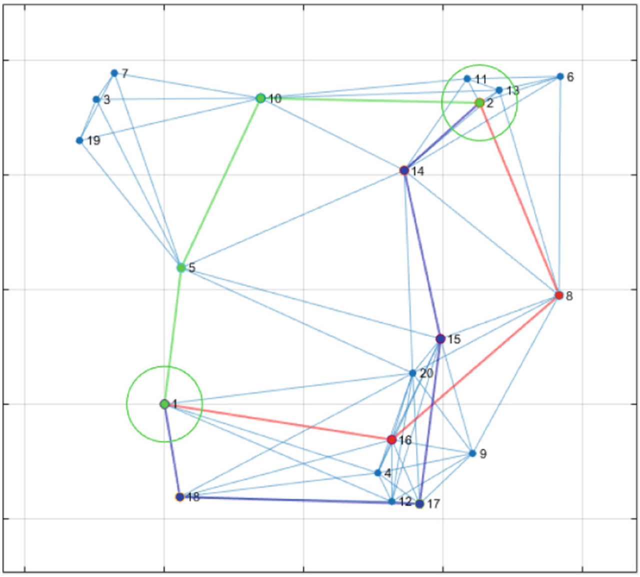

Fig. 5. Visualization of 3 selected routes by the fourth mechanism.

Obtained results confirm that same routes are very often selected by mechanisms 2 and 4. To illustrate the difference, note that between nodes 1 and 5 there was an error rate greater than 5% (that is, there was a probability that the packet would not arrive). Here, mechanism 4 took this situation into account and proposed rerouting packets. After this event, a change of three paths occurred (see, Table 5 and Fig. 6).

Table 5. Mechanism 4: the best three routes presented as consecutively visited nodes (after path change, due to link errors).

Lp	Node numbers on the path	Path cost	Number of intermediate nodes
1	[1, 16, 8, 2]	0.0994	4
2	[1, 18, 17, 15, 14, 2]	0.3099	6
3	[1, 20, 5, 10, 2]	0.6257	5

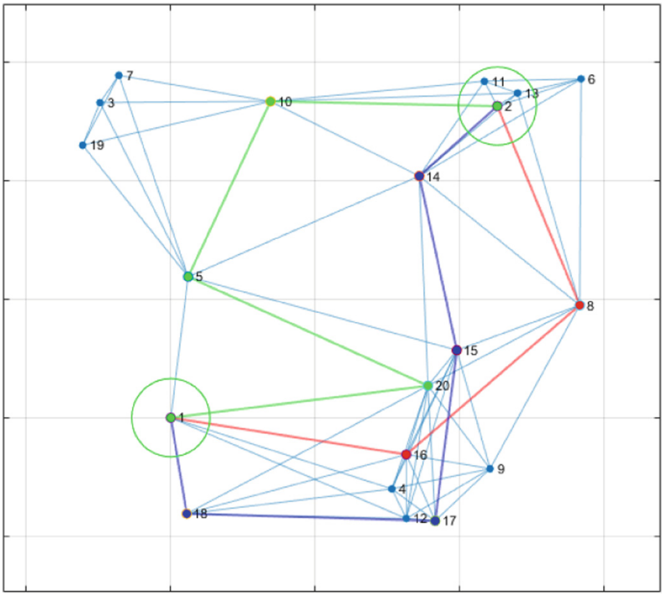

Fig. 6. Visualization of routes selected after path change, due to link errors.

5 Concluding Remarks

The results presented here are a contribution to the development of mechanisms for networks providing foundation of the Edge-Cloud Continuum ecosystems. The proposed solution deals with load balancing in wireless networks, targeting data collection and processing in distributed nodes. It applies betweenness measure, which allows establishing to what extent a node participates in the transmission of information. The adopted approach can be particularly useful for sensor infrastructure, with limited transmission capabilities and constraining energy consumption considerations. Here, reducing the risk of rejection of transmitted packets will increase the probability of successful data transfer, and will reduce delays. For this purpose, the extended mechanisms consider also link distance and edge error rate(s). This, among others, reduces the amount of energy consumed to increase the power of transmitters, to ensure communication between nodes that are far apart. Results of performed simulations indicate that, in the adopted solutions, paths based on edge nodes are very often selected, which root on avoiding central nodes that are heavily loaded with network traffic. In the future, further research is planned, related to the efficiency and reliability of load balancing mechanisms and the application to SDN/NFV environments.

Acknowledgement. This project is financed by the Minister of Education and Science of the Republic of Poland within the "Regional Initiative of Excellence" program for years 2019–2022. Project number 027/RID/2018/19, amount granted 11 999 900 PLN.

Work of Maria Ganzha, Marcin Paprzycki, Carlos Palau and Ignacio Lacalle Úbeda was sponsored in part by the ASSIST-IoT project, which received funding from the EU's Horizon 2020 research and innovation program under grant agreement No. 957258.

References

1. State of IoT 2021: Number of connected IoT devices growing 9% to 12.3 billion globally, cellular IoT now surpassing 2 billion. https://iot-analytics.com/number-connected-iot-devices/. Accessed 20 Dec 2021
2. Whitmore, A., Agarwal, A., Da Xu, L.: The internet of things—a survey of topics and trends. Inf. Syst. Front. **17**(2), 261–274 (2014). https://doi.org/10.1007/s10796-014-9489-2
3. Farahani, B., Firouzi, F., Chang, V., Badaroglu, M., Constant, N., Mankodiya, K.: Towards fog-driven IoT eHealth: promises and challenges of IoT in medicine and healthcare. Futur. Gener. Comput. Syst. **78**(2), 659–676 (2018)
4. Adasiewicz, R., et al.: Optimal placement of internet of things infrastructure in a smart building. In: Goyal, D., Gupta, A.K., Piuri, V., Ganzha, M., Paprzycki, M. (eds.) Proceedings of the Second International Conference on Information Management and Machine Intelligence. LNNS, vol. 166, pp. 661–669. Springer, Singapore (2021). https://doi.org/10.1007/978-981-15-9689-6_73
5. Jeschke, S., Brecher, C., Meisen, T., Özdemir, D., Eschert, T.: Industrial internet of things and cyber manufacturing systems. In: Jeschke, S., Brecher, C., Song, H., Rawat, D. (eds.) Industrial Internet of Things. Springer Series in Wireless Technology, pp. 3–19. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-42559-67_1
6. Nalinaksh, K., Lewandowski, P., Ganzha, M., Paprzycki, M., Pawłowski, W., Wasielewska-Michniewska, K.: Implementing autonomic internet of things ecosystems – practical considerations. In: Malyshkin, V. (ed.) PaCT 2021. LNCS, vol. 12942, pp. 420–433. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86359-3_32
7. Milojicic, D.: The edge-to-cloud continuum. Computer **53**(11), 16–25 (2020)
8. Vladyko, A., Khakimov, A., Muthanna, A., Ateya, A.A., Koucheryavy, A.: Distributed edge computing to assist ultra-low-latency VANET applications. Future Internet **11**(6), 128 (2019)
9. Jolfaei, A., Ostovari, P., Alazab, M., Gondal, I., Kant, K. Guest editorial special issue on privacy and security in distributed edge computing and evolving IoT. IEEE Internet Things J. **7**(4), 2496–2500 (2020)
10. Epstein, L.: Load balancing. In: Kao, M,Y. (ed.) Encyclopedia of Algorithms. Springer, Boston (2008)
11. Phillips, S., Westbrook, J.: On-line load balancing and network flow. Algorithmica **21**, 245–261 (1998)
12. Molina, B., Palau, C.E., Esteve, M.: Modeling content delivery networks and their performance. Comput. Commun. **27**(15), 1401–1411 (2004)

13. Aghdai, A., Wang, M.I.-C., Xu, Y., Wenz, C.H.-P., Chao, H.J.: In-network congestion-aware load balancing at transport layer. In: 2019 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), pp. 1–6. IEEE (2019)
14. López, C., Lacalle, I., Belsa, A., Kopertowski, Z., E Palau, C., Esteve, M.: Reviewing SDN adoption strategies for next generation internet of things networks. In: Somani, A.K., Mundra, A., Doss, R., Bhattacharya, S. (eds.) Smart Systems: Innovations in Computing. SIST, vol. 235, pp. 619–631. Springer, Singapore (2022). https://doi.org/10.1007/978-981-16-2877-1_57
15. Kavana, H.M., Kavya, V.B., Madhura, B., Neha, K.: Load balancing using SDN methodology. Int. J. Eng. Res. Technol. **7**(5), 206–208 (2018)
16. Babbar, H., Rani, S., Gupta, D., Aljahdali, H.M., Singh, A., Al-Turjman, F.: Load balancing algorithm on the immense scale of internet of things in SDN for smart cities. Sustainability **13**, 9587 (2021)
17. Junior, R.R.R., Vieira, M.A.M., Vieira, L.F.M., Loureiro, A.A.F.: Intra and inter-flow link aggregation in SDN. Telecommun. Syst. (2021) AQ2
18. Rajagopalan, S.: An Overview of Server Load Balancing. Int. J. Trend Res. Dev. **7**(2), 231–232 (2020)
19. Chen, W., Shang, Z., Tian, X., Li, H.: Dynamic server cluster load balancing in virtualization environment with OpenFlow. Int. J. Distrib. Sens. Netw. (2015)
20. Mishra, S.K., Sahoo, B., Parida, P.P.: Load balancing in cloud computing: a big picture. J. King Saud Univ. – Comput. Inf. Sci. **32**(2), 149–158 (2020)
21. Rahimi, P., Chrysostomou, C.: Improving the network lifetime and performance of wireless sensor networks for IoT applications based on fuzzy logic. In: 15th International Conference on Distributed Computing in Sensor Systems, pp. 667–674. IEEE (2019)
22. Hasan, M.Z., Al-Turjman, F.: Optimizing multipath routing with guaranteed fault tolerance in internet of things. IEEE Sens. J. **17**(19), 6463–6473 (2017)
23. Gupta, A., Christie, R., Manjula, R.: Scalability in internet of things: features. Int. J. Comput. Intell. Res. **13**(7), 1617–1627 (2017)
24. The Network as a Platform: Reliable Network. Scalable Networks. http://cisco.num.edu.mn/CCNA_R&S1/course/module1/1.3.2.4/1.3.2.4.html. Accessed 20 Dec 2021
25. Nakai, A., Madeira, E., Buzato, L.E.: On the use of resource. J. Netw. Syst. Manage. **23**, 502–538 (2015)
26. Li, L., Jin, Z., Li, G., Zheng, L., Wei, Q.: Modeling and analyzing the reliability and cost of service composition in the IoT: a probabilistic approach. In: 19th International Conference on Web Services, pp. 584–591. IEEE (2012)
27. Pourghebleh, B., Navimipour, N.J.: Towards efficient data collection mechanisms in the vehicular ad hoc networks. Int. J. Commun. Syst. **32**(5), 1–20 (2019)
28. AQ3 Tseng, Ch.H.: Multipath load balancing routing for internet of things. J. Sens. 1–8 (2016)
29. Bhandari, S., Sharma, S.K., Wang, X.: Latency minimization in wireless IoT using prioritized channel access and data aggregation. In: IEEE Global Communications Conference, pp. 1–6. IEEE (2017)
30. Ortiz, C., Puig, J., Palau, Carlos E. and Esteve, M.: 3D wireless sensor network modeling and simulation. In: International Conference on Sensor Technologies and Communications pp. 307–312 (2007)
31. Wan, J., Chen, B., Xia, M., Wang, S., Li, D., Liu, C.: Fog Computing for energy-aware load balancing and scheduling in smart factory. IEEE Trans. Industr. Inf. **14**(10), 4548–4556 (2018)
32. Thubert, P., et al.: RPL: IPv6 routing protocol for low-power and lossy networks RFC 6550 (2012)
33. Taghizadeh, S., Bobarshad, H., Elbiaze, H.: CLRPL: context-aware and load balancing RPL for IoT networks under heavy and highly dynamic load. IEEE Access **6**, 23277–23291 (2018)

34. Ghaleb, B., Al-Dubai, A., Elias, E., Gharibi, W., Mackenzie, L., Khalaf, M.B.: A new load-balancing aware objective function for RPL's IoT Networks. In: 20th IEEE Conference on High Performance Computing and Communications (HPCC-2018), pp. 909–914 (2018)
35. Hamrioui, S., Lorenz, P.: Load balancing algorithm for efficient and reliable IoT communications within E-health environment. In: GLOBECOM 2017 - 2017 IEEE Global Communications Conference, pp. 1–6 (2017)
36. Kotagi, V.J., Singh, F., Murthy, C.S.R.: Adaptive load balanced routing in heterogeneous IoT networks. In: 2017 IEEE International Conference on Communications Workshops (ICC Workshops), pp. 589–594 (2017)
37. Xiao, Y., Zhu, C.: Vehicular fog computing: vision and challenges. In: 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), pp. 6–9 (2017)
38. Hussain, F., Alkarkhi, A.: Big Data and Fog Computing. SpringerBriefs in Electrical and Computer Engineering. Springer, Cham (2017)
39. Grabowski, F., Paszkiewicz, A., Bolanowski, M.: Computer networks as complex systems in nonextensive approach. J. Appl. Comput. Sci. **21**(2), 31–44 (2013)
40. Menezes, R., Evsukoff, A., González, M.C.: Complex Networks. Springer, Heidelberg (2013)
41. Barthélemy, M.: Betweenness centrality in large complex networks. Eur. Phys. J. B **38**(2), 163–168 (2004). https://doi.org/10.1140/epjb/e2004-00111-4
42. Wi-Fi Efficiency – Poor Performance due Retransmission overhead. https://extremeportal.force.com/ExtrArticleDetail?an=000095772. Accessed 20 Dec 2021