Energy, QoS and Security Aware Edge Services

Erol Gelenbe[1,2](✉), Mateusz P. Nowak[1], Piotr Frohlich[1], Jerzy Fiolka[3], and Jacek Checinski[3]

[1] Institute of Theoretical and Applied Informatics, Polish Academy of Sciences,
Baltycka 5, 44-100 Gliwice, Poland
gelenbe.erol@orange.fr

[2] LabI3S, Universite Cote d'Azur, Grand Chateau, 06103 Nice, France

[3] Faculty of Automatic Control, Electronics and Computer Science, The Silesian
University of Technology, Akademicka 16, 44-100 Gliwice, Poland
http://www.iitis.pl

Abstract. With the development of communication technologies and the increasing bandwidth of optical fibres and transmission speeds in current 5G and future 6G wireless networks, there is a growing demand for solutions organising traffic in such networks, taking into account both end-to-end transmissions and the possibility of data processing by edge services. The most pressing problems of today's computer networks are not only bandwidth and transmission delays, but also security and energy consumption, which is becoming increasingly important in today's climate. This paper presents a solution based on neural networks that organises network traffic taking into account the above criteria - quality of service (QoS), energy consumption and security.

Keywords: SDN · Random Neural Networks · Green computing · Edge computing · Energy-awareness · Green networking · Security · IoT · QoS

1 Introduction

Today's communication technologies are capable of transmitting increasing amounts of data per second. Their source is not only the data of human-operated applications, but increasingly the sensors and hubs of major applications such as healthcare [6,31] and the of the Internet of Things (IoT) and other services. However, the Internet's ease of use and high bandwidth also creates tremendous opportunities for attackers, so that all these Internet accessible systems need to be protected from malicious attacks [5,32].

Since the computational capabilities of servers and workstations are limited and they are not always able to process data at an appropriate speed, Cloud

This work has been supported by the European Commission H2020 Program through the IoTAC Research and Innovation Action, under Grant Agreement No. 952684.

E. Gelenbe et al. (Eds.): EuroCybersec 2021, CCIS 1596, pp. 102–117, 2022.
https://doi.org/10.1007/978-3-031-09357-9_9

architectures have become the answer to this problem, grouping servers into structures that provide huge computing capacities, but these need to be properly accessed and scheduled [4,40,42]. The second trend, which is gaining momentum especially with the development of 5G networks, is the multiplication of computing services and their movement to the Edge, close to the users and to the sources of data.

The primary purpose of a computer system aand network is to process and transmit data while maintaining adequate Quality of Service (QoS) [20]. Disturbances in QoS result in the need to wait for data, thus wasting computing power, and often in the need to resend data, which in addition, in the case of IoT devices, is associated with energy expenditure and shortening the life of a battery-powered device. QoS problems could be avoided if it were possible to place processing nodes close enough to the data source so that transmission would not be a problem. However, this can be too costly, both at the investment stage and later when it comes to covering energy costs. Electricity, apart from being an obvious cost for the operator, is obtained in the overwhelming majority from non-renewable energy sources, and its unnecessary consumption has an impact on the climate of our planet. It should therefore be saved for both economic and ecological reasons [22]. How important, although underestimated, is 'green computing' and 'green networking' [3,35] is shown by the fact that, at present, the energy consumption of IT systems accounts for roughly 10% of global electricity consumption, and by 2030 this share may even reach 20% [1,16].

Another problem is security which needs to be assured [19,21]. As the value of data transmitted over the network and processed on external servers increases, so do the number of attacks on the infrastructure for transmitting, processing and storing information. Modern computer systems must take this issue into account already at the design level, according to the security-by-design principle.

Our ppaer addresses all these issues, improving network performance in terms of QoS, power consumption and security [27]. This article is composed of six main sections. In Sect. 2 we briefly introduce the reader to the topic of RNNs, referring to previous publications on the subject. We show the specifics of the environment that is the subject of the current research and the tailored solutions that we have used. Section ?? discusses how to collect QoS, energy and security data that RNNs use to make decisions. Section 3 presents the experimental part, including a description of the implementation and the testbed. It also includes a discussion of the obtained results. The whole work is summarised in the Sect. 4.

2 Random Neural Networks for the Control of Computer Networks

The optimization the QoS of distributed systems has been discussed in numerous publications [28,38,39,41,42]. QoS versus energy consumption of distributed services has also been examined experimentallly in [18]. However, the focus on security is more recent and its impact on network management and routing is examined in [10,11,17].

To control the network in terms of multiple criteria, including QoS, security and energy in our case, we use a solution based on Random Neural Networks (RNNs) [12,13], trained using Reinforcement Learning. RNNs optimize data packet transmission paths as well as the selection of Egde Computing services in such a way as to maintain an appropriate (predefined) balance between QoS, energy consumption and security. The switches and servers of a computer network form a distributed system, and its optimization is a variation of a well-known problem. However, by using the RNN and placing our system in a Software Defined Network (SDN) environment as in [8,9], we show that familiar Machine Learning techniques can also be used in state-of-the-art network architectures.

It should be noted, however, that the use of an SDN controller to implement the presented solutions is convenient from the point of view of demonstrating the usefulness of RNN in computer network control, but due to the distributed architecture of the RNN-based Decision Engine the same solutions can - under certain conditions - also be applied to a traditional, fully distributed network architecture.

The problems of SDN design and optimization are discussed in survey paper [36], taking into account not only energy efficiency issues, but also touching on security problems. Security issues in SDN have received a number of publications, for example in [2]. An interesting survey article on system deployment and optimization, shedding light on our work, was published in [25]. The popularity of this technology and the ease of implementation of routing control algorithms are also significant.

2.1 The Goal of the Decision System

The system we consider consists of:

- The set of network SDN switches or forwarders $S = \{s_1, \ .. \ s_n\}$ that are interconnected via a network graph, where S is the set of nodes and A is the $n \times n$ one-hop binary connection matrix between nodes.
- Every switch $s \in S$ may have connected "clients" or Edge services.
- The set of Clients is $C = \{c_1, \ ... \ c_m\}$ and each client c has a node or switch $s(c)$ to which it is directly connected .
- Edge services are used to offload specific cloud services (with their processing capacity and/or repositories) that are operating in close proximity so as to offer fast service to the clients. They belong to a set $E = \{e_1, \ ... \ e_M\}$ of M services which all offer equivalent facilities in terms of processing and the ability to provide specific data. Also any service e is connected to some switch or node $s(e)$.

The Goal of the decision system is to find a P among the set of switches S to connect the pair of clients $(c, c'), \ c, c' \in C$ or the client-service pair $(c, e), \ c \in$

$C,\ e \in E$. The choice of the path is based on the QoS, security and energy criteria, or one or two of these criteria. For ease of notation we will denote a connection (c, c') or $(, e)$ as a "flow" f.

Thus a path:

- $P = P(c, c')$ from c to c' is $P(c, c') = (s(c), s(P)_1,\ \dots .s(P)_{l(P)-2}, s(c'))$, or
- A path $P = P(c, e)$ from c to e is $P(c, e) = (s(c), s(P)_1,\ \dots .s(P)_{l(P)-2}, s(e))$, where
- $A(s(c), s(P)_1) = 1,\ A(s(P)_i, s(P)_{i+1}) = 1$, for $1 \le i \le l(P) - 3$, $A(P_{l(P)-2}, s(c')) = 1,\ A(P_{l(P)-2}, s(e)) = 1$,
- and $l(P)$ denotes the length of the path P in number of switches or nodes.

Thus we can now formulate the goal function G for given flow and path as the weighted sum of three criteria:

$$G(f, P) = aQ(f, P) + bT(f, P) + cJ(f, P), \tag{1}$$

where $a,\ b,\ c$ are non-negative constants with $a + b + c = 1$, and $Q(f, P)$ is the QoS value for given flow f using path P. For instance, $Q(f, P)$ can be the end-to-end delay per packet for flow f on path P or the corresponding packet loss, or some combination thereof. The measurement of such metrics is presented in Section ?? below.

$T(f, P)$ is the trust metric that expresses the level of insecurity of traffic belonging to given flow f going along the path P. It can be obtained via Attack or Anomaly Detectors, Honeypots or similar entities, that asseses the probability or some other non-negative metric, that connection f is harmed by devices on path P. Note that $T(f, P)$ may be symmetric so that it may characterize the effect of f on P, rather than the opposite. Furthermore it may be expressed as the cumulative effect of all the nodes on path P, such as:

$$T(f, P) = \sum_{s \in P} T(f, s),\ \textit{or}\ T(f, P) = \max\{T(f, s) :\ s \in P\}. \tag{2}$$

$J(f, P)$ is the energy consumed per packet by flow f by devices along path P, which can be computed from the power consumption and traffic rate, as follows:

$$J(f, P) = \sum_{s \in P} \frac{\Pi(s, \lambda(s))}{\lambda(s)}, \tag{3}$$

where $\Pi(s)$ is the power consumption when switch or node s carries the traffic rate $\lambda(s)$ while:

$$\lambda(s) = \sum_{f \in F} \sum_{s \in f} \lambda(f), \tag{4}$$

and $\lambda(f)$ is the traffic rate of connection f, and $F = \{f\}$ is the set of all active connections.

2.2 RNN Based Routing for Path Control

The approach taken here is to use the Cognitive Packet Network (CPN) idea [14,15,23], so as to store inside the SDN Controller a "good" or near-optimal path $P(f)$ for flow $f = (c, e)$ from client c to edge device e that minimizes $G(f, P(f))$. Thus, rather than calculate ex-nihilo for each upcoming connection $f = (c, e)$ the path $P(f)$, we follow the CPN approach that maintains for each router or switch (i.e. node) s, a Random Neural Network [12] that computes the best "next hop" from s to $s'(s, e))$, where $s'(s, e)$ is the node to which s is connected and that minimizes $G((s, e), P(s, e))$.

Since our study is focused on the IoT where the real-time operation is crucial, the path link latencies were chosen as the key QoS metric. Since a SDN controller within its standard means has no direct way to measure the latency on the links and paths, Cognitive Packets (CP) were employed as described in [24] were described for this purpose. CPs have also been employed in SDN networks previously [9,33,34], but the concept of the Cognitive Network Map (CNM) was extended with all necessary data within single data structure.

2.3 Energy

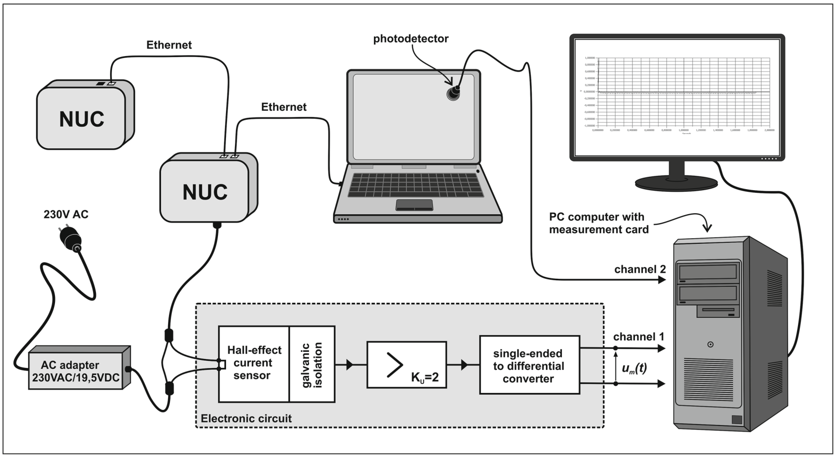

Fig. 1. Measurement circuit for power versus traffic characteristics.

Most network devices do not have the ability to directly measure energy during operation. However, since each network packet handled needs to be processed and transmitted, it is obvious that the amount of energy consumed during operation of a network switch depends on the traffic intensity. The energy characteristic reflecting the amount of energy in Watts [W] depending on the amount of network traffic passing through the switch is, on the one hand, easy to measure in the laboratory, and on the other hand - during operation in the real system - gives the SDN controller, knowing the current throughput of the node, a sufficiently precise answer to the question "How much energy does the network switch consume at this moment".

The SDN switches used in our experiments are Intel NUC devices [26] that run Open vSwitch [29]. Our approach, however, is universal in the sense that it can be applied to any network switch or router.

The laboratory setup used for the measurements if the power drawn during data transfer is presented in Fig. 1. After setting of the traffic level given in Mb/s the energy measurement was done. The traffic was generated and received by workstations connected to the NUC device. The experiment was carried out for successive for increasing traffic levels as shown in Fig. 2.

The electronic circuit which is used to condition the signal obtained from a sensor which measures the current, is based on precision operational amplifiers. The Hall effect-based current sensor ACS712-05 (0–5A current range) is galvanically isolated from the copper conduction path, integrated into the IC, which is used to pass the measured current. This path was connected in series with the supply wire on the constant DC voltage side at $U_{DC} = 19.5V$, of the AC adapter used for the NUC's as shown in Fig. 1. The output signal from the sensor is amplified in a single-ended amplifier and then converted to the differential form. The instantaneous value of the measured power can then be found from the following relationship:

$$P = U_{DC}.i = U_{DC}\frac{U_m}{k_u S} = AU_m,\ in\ Watts\ , \tag{5}$$

where $S = 185mV/A$ is the sensitivity of the current sensor, and $A = U_{DC}/(k_u S) = 520.9A$ is a constant with $k_u = 2$ which is related to the instrumentation, and U_m is the measured output voltage of the single-sided differential converter shown at "channel 1" of Fig. 1, which results from the Hall-effect measurement of the NUC input current.

To reduce the effect of noise and interference, thirty separate measurements were repeated for the power consumption as a function of incoming and outgoing traffic, and the results are summarised in Fig. 2. Then we extracted the difference of the energy consumption between the basic level for zero traffic and the value for a given traffic level, and the increase of energy consumption per traffic volume in Mb is presented in Fig. 3.

2.4 Security

The level of trust in a given flow, and therefore in the device that generates it, can be assessed using external entities. Within the network, nodes and devices with higher and lower sensitivity may be defined. For example, the failure of some nodes has a greater impact on the operation of the entire network than in the case of other nodes, and attacking such a node will cause more damage than otherwise. Security-aware routing aims to direct suspicious traffic away from

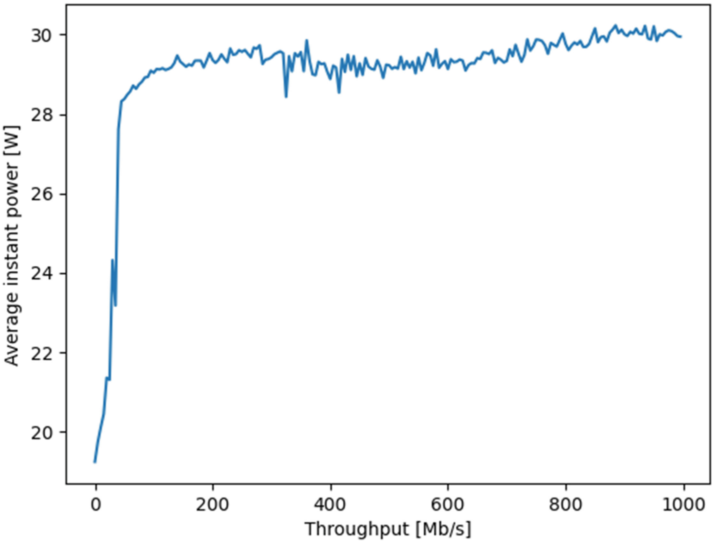

Fig. 2. The dependence between the instantaneous power consumption and traffic load of the Intel NUC when used as a switch or router.

vulnerable nodes, if possible. Trust assessing entities can be Attack Detectors or Honeypots, e.g. [17,30]. We employed SYN attack detector presented in [7].

3 Experiments and Results

The experiments we performed were done in the IITiS laboratory. The test network consisted of seven NUC devices working as SDN switches, plus SDN controller, client machines and attack detector. The basic topology of the network is presented in Fig. 4

For clarity of results presentation, and in order to concisely present the different possibilities of our solution, two separate experiments were performed, however the basic network configuration remained the same. The course and results of the experiments follows.

3.1 Point-to-Point Transmission in Insecure Environment

The aim of the experiment was to reflect the situation of point-to-point communication in the situation of an attack. As presented in Fig. 6, point-to-point communication from c_1 to c_6 client devices was established and put under observation. In this experiment energy efficiency was not taken into consideration, to avoid too many factors influencing the results, making it hard to separate the influence of each of them on the final results.

The experiment had three steps:

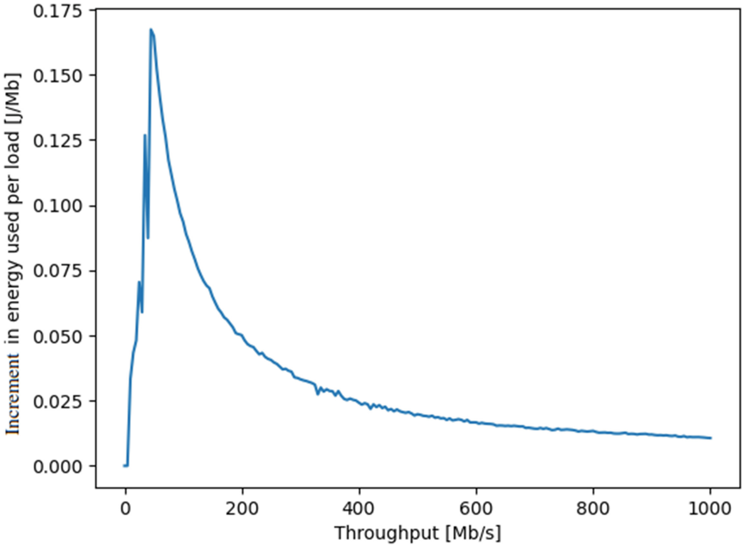

Fig. 3. The energy used per Mb in the function of switch load.

- Normal communication from c_1 to c_6
- QoS deterioration on the link c_1–c_4
- Security problem detected – the need to bypass sensitive nodes c_3 and c_5

The measurement included latency on the path c_1 to c_6. System reaction to changing conditions can be easily observed in the Fig. 5. After some time needed for the neural network to test various conditions and possibilities the path which is both fast and secure was found. The network configurations in particular steps are presented in Figs. 6, 8 and the final on in 8

3.2 Energy-efficient Access to the Edge

The final topology of the second experiment is presented in Fig. 9. It include 24 client devices (implemented as virtual machines) and seven edge services. Every switch was accompanied by the separate service instance. The energy characteristics is taken into account, as well as total time of request handling by the Edge services. The total handling time included: time of client-to-service communication t_{cs}, request handling in the server t_r, time of service-to-client communication t_{sc}. The second component of the goal function was energy efficiency, and energy characteristics from the Fig. 3 was loaded into SDN controller for readouts of energy usage based on traffic in each switch. The RNN decision engine was used for path-and-service choice (Fig. 7).

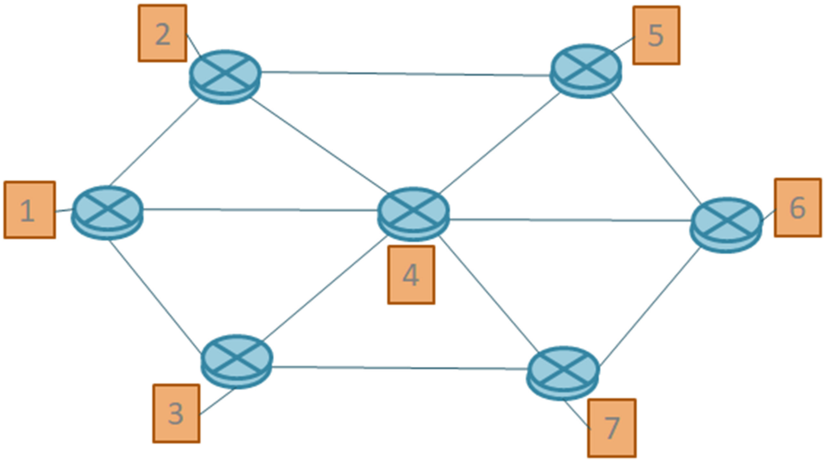

Fig. 4. The configuration of experimental test-bed

The course of the experiment included loading the network with heavy traffic of stress-test type, as such a load was best to show differences in energy usage. Seven steps of experiments were performed, in every step the total load in the network was increased by 1 Gb/s. In the first run only QoS optimisation was performed as a reference result, then both QoS and Energy components were included into the Goal functions. The results, presented in Figs. 10 and 11, show positive influence of the latter version of Goal function on the total energy consumption. with minor effect on QoS.

4 Conclusions

The paper presents the possibilities of using modern tools from the field of Artificial Intelligence (AI) and Machine Learning (ML) to control the operation of computer networks. It has been shown that theoretical capabilities of RNNs can be translated into practical applications, and appropriately constructed goal functions perform complex routing based on several criteria simultaneously.

Among the criteria tested experimentally are the possibilities of increasing the security and reducing the energy consumption of the IT infrastructure, which are very relevant for today's IT systems. These very promising ideas have been tested in several experiments which demonstrate their practical value in the framework of Software Defined Networks.

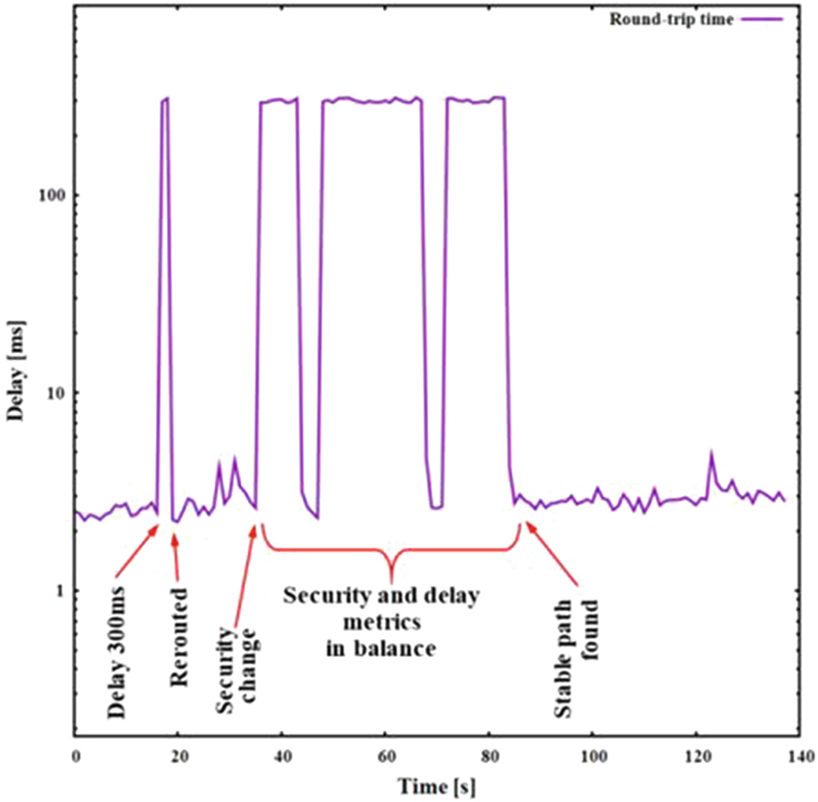

Fig. 5. The delay in time between clients 1 and 6

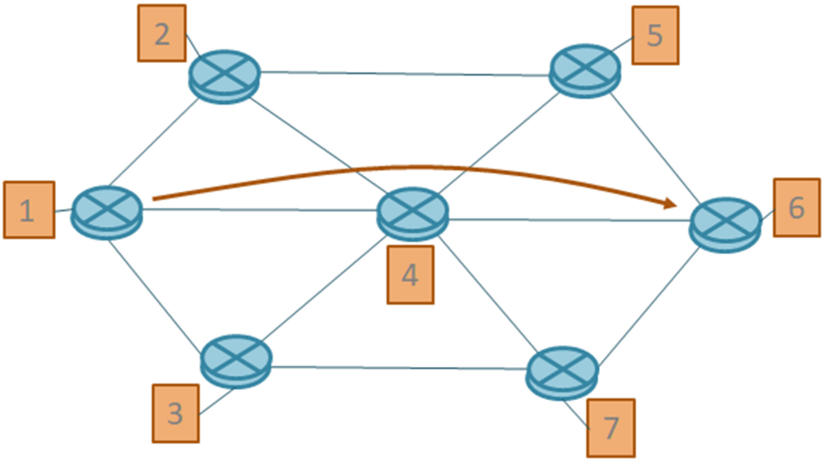

Fig. 6. The c_1–c_6 path configuration – stage 1

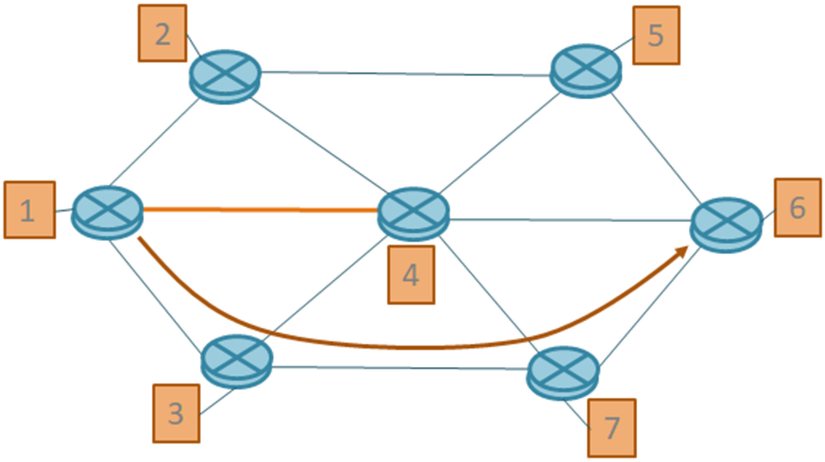

Fig. 7. The c_1–c_6 path configuration – stage 2

References

1. Andrae, A.S.G., Edler, T.: On global electricity usage of communication technology: trends to 2030. Challenges **6**(1), 117–157 (2015)
2. Aytaç, S., Ermiş, O., Çağlayan, M.U., Alagöz, F.: Authenticated quality of service aware routing in software defined networks. In: Zemmari, A., Mosbah, M., Cuppens-Boulahia, N., Cuppens, F. (eds.) CRiSIS 2018. LNCS, vol. 11391, pp. 110–127. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-12143-3_10
3. Berl, A., et al.: Energy-efficient cloud computing. Comput. J. **53**(7), 1045–1051 (2010)
4. Brun, O., Wang, L., Gelenbe, E.: Big data for autonomic intercontinental overlays. IEEE J. Sel. Areas Commun. **34**(3), 575–583 (2016)
5. Brun, O., Yin, Y., Gelenbe, E.: Deep learning with dense random neural network for detecting attacks against IoT-connected home environments. Procedia Comput. Sci. **134**, 458–463 (2018)
6. Diamantopoulos, S., et al.: Secure cross-border exchange of health related data: the KONFIDO approach. In: Montella, R., Ciaramella, A., Fortino, G., Guerrieri, A., Liotta, A. (eds.) IDCS 2019. LNCS, vol. 11874, pp. 318–327. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-34914-1_30
7. Evmorfos, S., Vlachodimitropoulos, G., Bakalos, N., Gelenbe, E.: Neural network architectures for the detection of SYN flood attacks in IoT systems. In: PETRA 2020: Proceedings of the 13th ACM International Conference on PErvasive Technologies Related to Assistive Environments. Association for Computing Machinery, New York, NY, United States, Corfu, Greece, June 2020. https://doi.org/10.1145/3389189.3398000
8. Francois, F., Gelenbe, E.: Optimizing secure SDN-enabled inter-data centre overlay networks through cognitive routing. In: 2016 IEEE 24th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS), pp. 283–288. IEEE (2016)

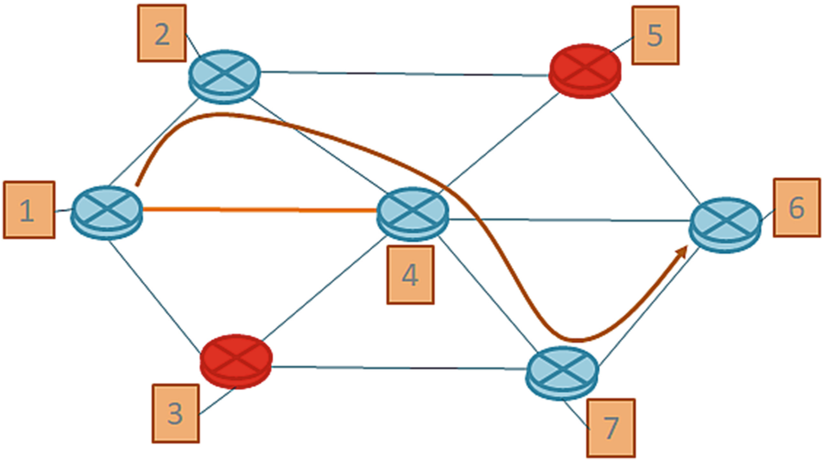

Fig. 8. The c_1–c_6 path configuration – stage 3

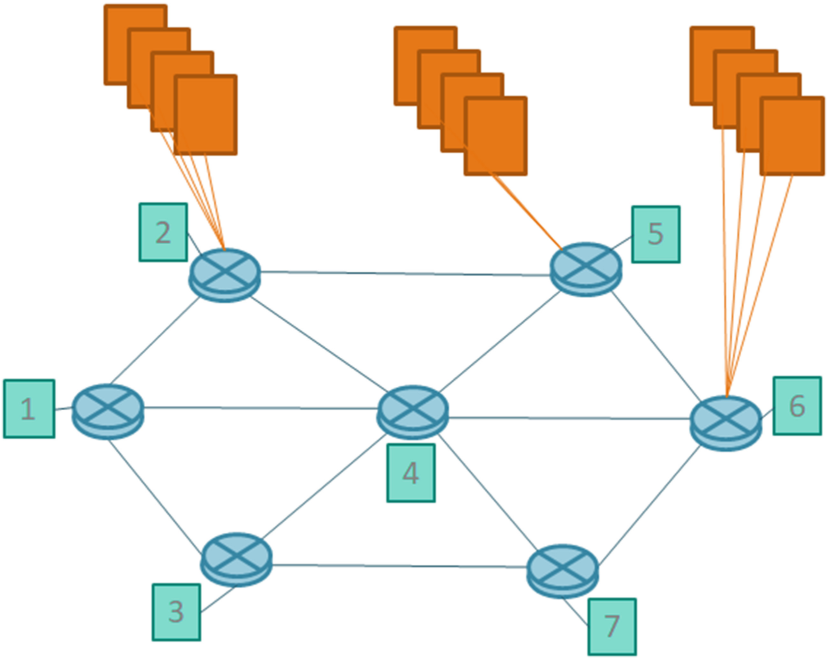

Fig. 9. Configuration of the Edge services experiment

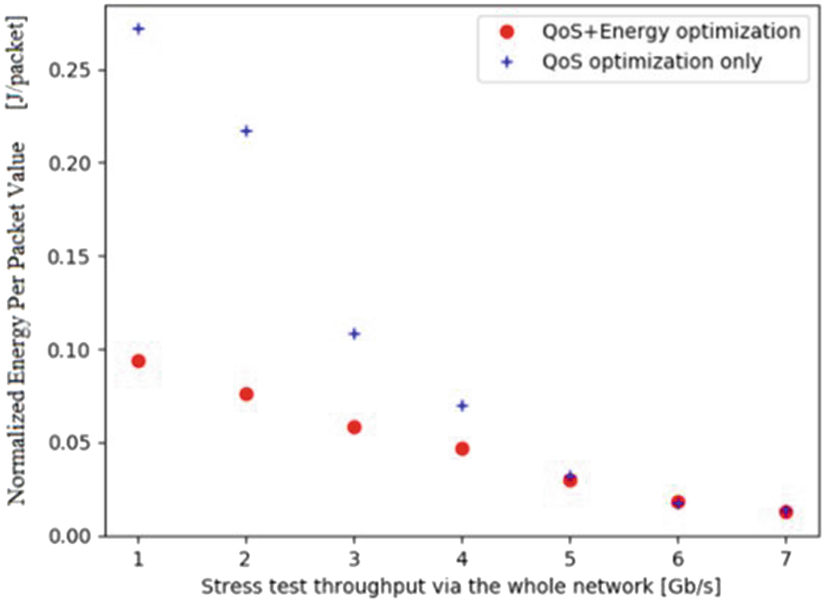

Fig. 10. Average Energy [J]/packet during stress test

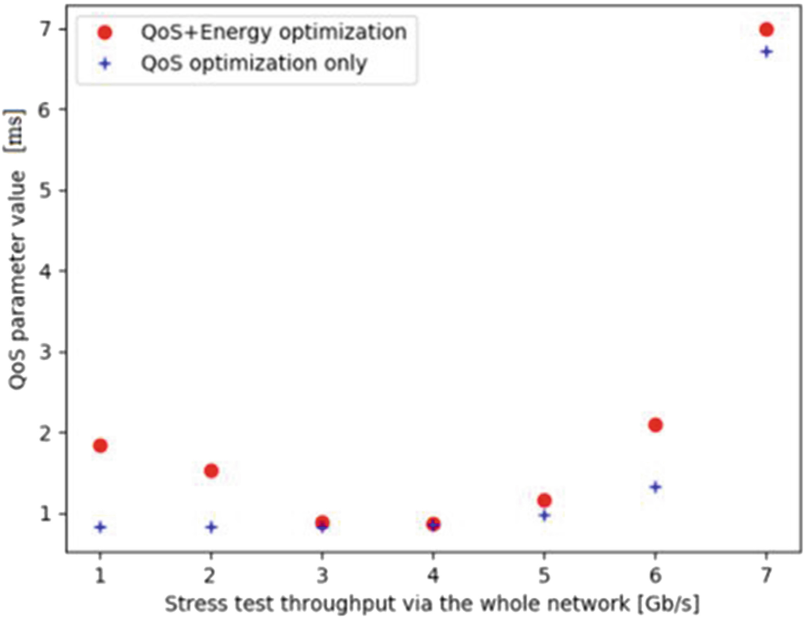

Fig. 11. Average QoS (delay [ms]) during stress test

9. François, F., Gelenbe, E.: Towards a cognitive routing engine for software defined networks. In: 2016 IEEE International Conference on Communications (ICC), Kuala Lumpur, May 22–27, pp. 1–6 (2016). https://doi.org/10.1109/ICC.2016.7511138
10. Frohlich, P., Gelenbe, E., Nowak, M.P.: Smart SDN management of fog services. In: GIOTS 2020: Global IoT Summit 2020, IEEE Communications Society, 1–5 June 2020, Dubin, Ireland. TechRxiv (2020)
11. Fröhlich, P., Gelenbe, E., Nowak, M.P.: Smart SDN management of fog services. In: 2020 Global Internet of Things Summit (GIoTS), pp. 1–6 (2020). https://doi.org/10.1109/GIOTS49054.2020.9119542
12. Gelenbe, E.: Random neural networks with negative and positive signals and product form solution. Neural Comput. **1**(4), 502–510 (1989)
13. Gelenbe, E.: Learning in the recurrent random neural network. Neural Comput. **5**(1), 154–164 (1993)
14. Gelenbe, E.: Cognitive Packet Network. US Patent US6804201B1 (2004)
15. Gelenbe, E.: Steps toward self-aware networks. Commun. ACM **52**(7), 66–75 (2009)
16. Gelenbe, E., Caseau, Y.: The impact of information technology on energy consumption and carbon emissions. Ubiquity **2015**, 1–15 (2015)
17. Gelenbe, E., Fröhlich, P., Nowak, M., Papadopoulos, S., Protogerou, A., Drosou, A., Tzovaras, D.: IoT network attack detection and mitigation. In: 9th Mediterranean Conference on Embedded Computing (MECO), Budva, Montenegro, pp. 1–6 (2020). https://doi.org/10.1109/MECO49872.2020.9134241
18. Gelenbe, E., Lent, R.: Energy-QoS trade-offs in mobile service selection. Future Internet **5**(2), 128–139 (2013). https://doi.org/10.3390/fi5020128
19. Gelenbe, E., Loukas, G.: A self-aware approach to denial of service defence. Comput. Netw. **51**(5), 1299–1314 (2007)
20. Gelenbe, E., Mitrani, I.: Analysis and Synthesis of Computer Systems, vol. 4. World Scientific, London (2010)
21. Gelenbe, E., Pavloski, M.: Performance of a security control scheme for a health data exchange system. In: IEEE International Black Sea Conference on Communications and Networking 26–29 May 2020 // Virtual Conference, pp. 1–6. IEEE (2020)
22. Gelenbe, E., Siavvas, M.: Minimizing energy and computation in long-running software. Appl. Sci. **11**(3), 1169 (2021)
23. Gelenbe, E., Xu, Z., Seref, E.: Cognitive packet networks. In: Proceedings 11th International Conference on Tools with Artificial Intelligence, Chicago, IL, USA, pp. 47–54 (1999). https://doi.org/10.1109/TAI.1999.809765
24. Gelenbe, E., Xu, Z., Seref, E.: Cognitive packet networks. In: Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence, pp. 47. ICTAI 1999, IEEE Computer Society, USA (1999)
25. Huang, X., Cheng, S., Cao, K., Cong, P., Wei, T., Hu, S.: A survey of deployment solutions and optimization strategies for hybrid SDN networks. IEEE Commun. Surv. Tutorials **21**(2), 1483–1507 (2019). https://doi.org/10.1109/COMST.2018.2871061
26. Intel: NUC - Small Form Factor Mini PC. https://en.wikipedia.org/wiki/Next-Unit-of-Computing (2021)
27. Kehagias, D., Jankovic, M., Siavvas, M., Gelenbe, E.: Investigating the interaction between energy consumption, quality of service, reliability, security, and maintainability of computer systems and networks. SN Comput. Sci **2**(1), 1–6 (2021)
28. Kim, C., Kameda, H.: An algorithm for optimal static load balancing in distributed computer systems. IEEE Trans. Comput. **41**, 381–384 (1992)

29. McKeown, N., et al.: OpenFlow: enabling innovation in campus networks. ACM SIGCOMM Comput. Commun. Rev. **38**(2), 69–74 (2008)
30. Nakip, M., Gelenbe, E.: MIRAI botnet attack detection with auto-associative dense random neural network. In: 2021 IEEE Global Communications Conference. Barcelona, Spain, December 2021
31. Nalin, M., et al.: The European cross-border health data exchange roadmap: case study in the Italian setting. J. Biomed. Inf. **94**, 103183 (2019)
32. Natsiavas, P., et al.: Developing an infrastructure for secure patient summary exchange in the EU context: Lessons learned from the konfido project. Health Inf. J. **27**(2), 14604582211021460 (2021)
33. Nowak, M., Nowak, S., Domanska, J.: Cognitive routing for improvement of IoT security. In: Proceedings of IEEE International Conference on Fog Computing ICFC, Prague (2019). https://doi.org/10.13140/RG.2.2.28667.36648
34. Nowak, M., Nowak, S., Domańska, J., Czachórski, T.: Cognitive packet networks for the secure internet of things. In: Global IoT Summit (GIoTS). Aarhus, Denmark (2019)
35. Pernici, B., Aiello, M., Vom Brocke, J., Donnellan, B., Gelenbe, E., Kretsis, M.: What is can do for environmental sustainability: a report from caise'11 panel on green and sustainable is. Commun. Assoc. Inf. Syst. **30**(1), 18 (2012)
36. Rawat, D.B., Lenkala, S.R.: Software defined networking architecture, security and energy efficiency: a survey. IEEE Commun. Surv. Tutorials **19**(1), 325–346 (2017). https://doi.org/10.1109/COMST.2016.2618874
37. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. 2nd Ed. MIT Press, Cambridge (2018)
38. Tian, W., Zhao, Y., Zhong, Y., Xu, M., Jing, C.: A dynamic and integrated load-balancing scheduling algorithm for cloud datacenters. In: Proceedings of IEEE International Conference on Cloud Computing and Intelligence Systems, pp. 311–315 (2011)
39. Topcuouglu, H., Hariri, S., Wu, M.Y.: Performance-effective and low-complexity task scheduling for heterogeneous computing. IEEE Trans. Parallel Distrib. Syst. **13**(3), 260–274 (2002)
40. Wang, L., Brun, O., Gelenbe, E.: Adaptive workload distribution for local and remote clouds. In: 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 3984–3988 (2016)
41. Zhang, Z., Zhang, X.: A load balancing mechanism based on ant colony and complex network theory in open cloud computing federation. In: Proceedings of 2nd International Conference Industrial Mechatronics Automation, vol. 2, pp. 240–243 (2010)
42. Zhu, X., Qin, X., Qiu, M.: Qos-aware fault-tolerant scheduling for real-time tasks on heterogeneous clusters. IEEE Trans. Comput. **60**(6), 800–812 (2011)