An Enhanced Software Framework for Improving QoS in IoT

Uma Tomer
Faculty of Computer Applications
Manav Rachna International Institute of Research and Studies, Faridabad
Haryana, India
uma.tomer@gmail.com

Parul Gandhi
Faculty of Computer Applications
Manav Rachna International Institute of Research and Studies, Faridabad
Haryana, India
parul.fca@mriu.edu.in

Received: 20 May 2022 | Revised: 25 June 2022 | Accepted: 1 July 2022

Abstract-Internet of Things (IoT) and Artificial Intelligence (AI) with its subcomponents are the latest emerging technologies that make our daily lives easier. Quality of Service (QoS) plays a very important role in IoT due to the large number of interconnected nodes. QoS is inversely dependent on the node count, i.e. the increment of nodes causes hampering to QoS, as increasing the number of nodes increases the number of requests to the IoT server. An enhanced framework is strongly needed to control QoS in IoT applications. This study proposes and implements an enhanced framework using Matlab, to control the number of requests. The proposed model can improve QoS parameters like throughput, latency, and packet loss by reducing the number of requests generated by the end nodes without compromising the services to the end user. The results showed that QoS parameters improved in terms of throughput by 5-10%, packet loss by up to 6%, and packet latency by 4%. This model can also be tested in hardware and may provide a better QoS solution.

Keywords-QoS; packet loss; latency; throughput; QoL; PoE

I. INTRODUCTION

Technology is transforming daily life and the world is getting more and more digitized, as objects are interconnected to the internet to provide. Digital applications exploit networks, wireless sensors, Wireless Local Area Networks (WLANs), and the Internet to improve their performance. Everything related to IoT can be considered as an edge node that helps in sensing, measuring, and interpreting data, and therefore requires good internet connectivity [1-5]. The huge number of complex interactions between devices in edge nodes causes difficulties in achieving dynamic QoS requirements. Service-oriented IoT is based on dynamic services for various applications and software tools to process and monitor parameters. This technology faces many challenges such as power consumption, lack of compliance, botnet attack, real-time sensing, and failure of components [1]. Various architectures have been proposed for traditional networks, but the existing traditional QoS attributes are inappropriate, and more QoS attributes should be considered to include energy consumption, information accuracy, better utilization of the network, and IoT coverage [6]. To overcome these problems, a new QoS model is required to improve and balance data accuracy and the quality of the data delivered to the sensors [6]. The accuracy of the information is the key to closing the gap between sensor data and the actual world [6].

IoT technology faces challenges such as standardization, cyber security, and energy management [6-8]. These problems motivated researchers to examine new IoT architectures for QoS, considering traditional and special attributes such as accuracy, energy consumption, and network optimization. This study proposes a three-layer architecture, i.e. sensor, network, and architecture, to achieve adequate performance depending on the demanded service. The Building Management System (BMS) is responsible for automatic regulation, control, and maintenance of predefined parameters to control functionality. It improves the Quality of Life (QoL) by improving the management system including transportation, smart grids, traffic lights, surveillance, and smart services. IoT-based systems can support these requirements easily by using Power over Ethernet (PoE), which offers an opportunity to revolutionize these devices. A survey study showed that buildings are responsible for consuming around 40% of the total energy consumption, but the BMS is increasing rapidly due to IoT applications. This study aims to:

- Investigate and analyze the main parameters that are vital for improving the performance of IoT QoS.
- Propose a framework to improve certain parameters of the IoT network, such as latency, throughput, and packet loss.
- Propose a novel algorithm to reduce the number of requests generated by IoT end-nodes and hence reduce the load of the IoT server.

II. RELATED WORKS

Although QoS for traditional networks has been explored a lot, the research on service-oriented IoT was comparatively less. A three-layered architecture was proposed for service-oriented IoT in [1], including the Application, Network, and Sensing Layers, to optimize scheduling performance and minimize resource costs. The Application Layer was used to explore optimal QoS-aware services using component services.

Corresponding author: Uma Tomer

The Network layer was used to deal with the scheduling of heterogeneous network environments. The Sensing Layer was used to deal with information acquisition and resource allocation. Edge Computing improved the user experience by bringing computing resources closer to the location where IoT produces data [7]. IoT users face QoS hindrances for the isolated execution of their applications. The skillful pairing of cloudlets to IoT applications is the primary task to resolve QoS constraints. A bilateral solution for edge services was proposed in [7], taking into account the demands of QoS in terms of service response time and establishing the dynamic pricing of the edge service rooted in the motives of cloudlets, IoT users, and the system.

The existing energy management mechanism fails to estimate real-time computation and the mobility of battery-powered IoT [8]. This study presented real-time computations under QoS constraints for battery applications and proposed a mobility-aware network for lifetime maximization. This was performed in two stages, online and offline. The online stage included a time-efficient QoS for the execution of a task to the frequently changing QoS requirements, while optimal mobility maximized network lifetime in the offline stage. A QoS-constrained IoT system operating with Finite Block Length (FBL) codes was proposed in [9] to support low latency communications, studied the arrival model and the deadline limit, and presented a QoS-constrained throughput expression. This study proposed an optimal power control algorithm to maximize throughput while guaranteeing a certain reliability target.

Despite the enormous attempts in standardization to reach the full potential of IoT, many challenges still exist. In [10], the QoS parameters and metrics were highlighted to improve an IoT device, considering QoS metrics such as throughput, delay, and packet loss. As multimedia services demand equalized QoS, they can use Quality of Experience (QoE) to dynamically assign resources [11]. This study determined various methods to enhance QoE and presented a mathematical model to meet the desired QoS. As QoS factors have a strong influence on QoE, IoT services were examined by equating three fundamental metrics of QoS: small loss, latency, and jitter [12]. These studies show that there is room for further improvement in the QoS of IoT. As IoT applications are vital in different parts of life, such as health monitoring [13, 14] or intrusion detection [15], there is still a need to improve existing algorithms and improve the throughput, latency, packet loss, and performance of the IoT server by reducing the number of packets.

III. PERFORMANCE PARAMETERS FOR QOS OF IOT

IoT is expanding rapidly in wireless communications [16]. Cisco proposed a standard IoT framework having 7 IoT levels, distributed in 3 computational layers: Edge-side, Server/cloud-side, and User-side layer. These computational layers transform real-world data into application visions. The Edge-side layer consists of 3 different levels in the framework: Edge nodes, Communication, and Edge Computing. Edge nodes provide the intelligence to sense, measure, and connect the Internet gateway to the cloud [18]. IoT Information Communication Technology (ICT) transfers information human to human, human to things, and things to things [19]. Edge-Computing is an emerging model that extends the cloud and its services to the edge of the network [20]. The Server/Cloud-side layer embeds data accumulation and abstraction, requiring high computational power, transmission cost, and delay [16]. To address the needs of non-real-time applications, real-time data is acquired in the data aggregation stage, which determines whether the data is relevant or not for the required application. The data abstraction stage conducts data preparation for the consumer application. The User-side layer consists of users and centers.

The performance parameters of QoS in IoT are latency, accuracy, and Packet Error Rate (PER). Latency is a measure of delay. In networking, latency is the measurement of data transmission to the target. In [21], the latencies of 3 Amazon Web Service (AWS)_ regions EAST-1, EAST-2, and WEST-2, and three Azure areas WEST, CENTRAL, and EAST were examined. To handle the latency of IoT instruments, it was noted that the aggregated processing latency in AWS was more than in Azure. The average latency was 45, 49, and 46ms for AWS EAST-1, EAST-2, and WEST-2 respectively, while 14, 12, and 4ms were the latencies for Azure WEST, CENTRAL, and EAST respectively. In [22], the analytical and downlink latency was compared, deriving expressions and highlighting performance tradeoffs in channel scheduling. The simulator was developed in Matlab and it was observed that it had a latency of 10ms.

Accuracy measures how close the displayed measurement is to the actual value of a signal. A framework was proposed in [23] to evaluate the accuracy by estimating the number of disk I/O per process. It was implemented with the INU emulator in Matlab by fixing 2 parameters: n as the number of buckets in the bucket array, and m as the number of buckets in each bucket subarray. Three simulations were performed for t= 1, 5, and 10 minutes, and n and m were observed to be 20 and 5 respectively, to achieve accuracy. Piggybacking was used in [24] to improve accuracy. A comparison of the Shewhart and Least Mean Square (LMS) methods was performed for data-transmission reduction of the two datasets. For the indoor scenario, the numbers of transmissions were 408 and 58 for LMS and Shewhart respectively. For the outdoor scenario, the number of transmissions was 682 for LMS and 201 for Shewhart. During the measurements, it was observed that the current consumption increased with the increase in piggyback, and the battery life reduced.

PER is used to test the access terminal's receiver performance. A newscast approach was proposed in [25], using a moving average along with a network of devices to survey the PER of the current frame. This process can increase the probability of data renewal by providing the aimed number of encrypted yield packets to meet the number of required packets for appropriate data decoding and recovery. When using the legacy procedure, the data recovery probability strongly decreases accompanying the increase in packet inaccuracy. The packet error probability was between 0.1 and 0.3, and the PER was 0.012 for the legacy and 0.015 for the proposed model. PER in Wireless Sensor Network (WSN) for Heating, Ventilation, and Air Conditioning (HVAC) systems was investigated in [26]. PER data were captured for 1 month. Temperature and humidity were collected and packet loss percentage was reported. The data were split into 3 classes: temperature and moisture (Class 1), neighbor node ID along with the number of hops (Class 2), amount of packets and small rebroadcast counts (Class 3). It was noticed that impenetrable node classification regionally can considerably reduce packet drain. The packet loss ratio of Class 1 was higher compared to the other two classes, possibly due to the large size of the packets. It was also deduced that a network could be simulated to decrease the leap counts by balancing the optimum count of nodes and separation.

IV. PROPOSED FRAMEWORK

An IoT network consists of a large number of interconnected devices. Some interconnections are based on local networking, and some of them require an Internet connection. The QoS of the IoT is based on the requests generated per unit of time for a particular IoT platform. An IoT platform can use Bluetooth, WiFi, Zigbee [27], or satellite networks [28]. Figure 1 shows the conventional and the proposed architecture of the IoT layer and the generation of a request to the IoT server. If the number of requests for a particular application can be reduced, then the QoS can be enhanced. There are many IoT platforms available on the market that focus on enhancing the devices to be served. The mutual benefit of an IoT hardware manufacturer and an IoT platform developer is that the more devices demanded, the more profit can be earned. But there is a very good scope for enhancing the QoS of a particular server by reducing its load.

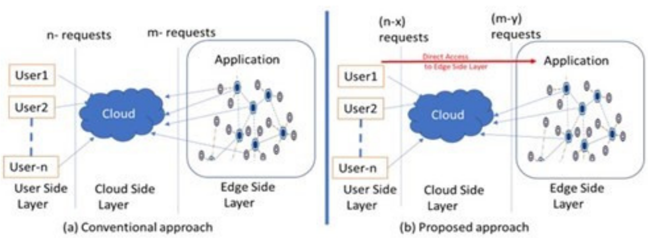

Fig. 1. (a) Conventional and (b) proposed IoT framework.

The proposed framework is based on a BMS application. Increasing the number of IoT devices requires a greater number of IoT devices connected to the IoT cloud, increasing the load of the server and compromising QoS as a greater number of requests is generated. Figure 1(a) shows the conventional approach of an IoT framework where both the user and the edge side layer generate requests to access the Internet. This may be to access the devices, log data to the cloud, etc. This generates a higher number of requests to the IoT service provider. Figure 1(b) shows the proposed approach of the IoT framework, which is based on user requirements. There are many tasks where the user is near to the edge side layer, so he can directly access the data from the edge side layer. In such cases, low-distance wireless communication can help a lot. Wi-Fi, Bluetooth, and Zigbee are some of the protocols which can be accessed locally, so some requests generated by this model could be handled locally by the user. This approach can significantly enhance the QoS by decreasing the load over the IoT service platform. Figure 1(b) also shows that the number of requests generated by the edge side layer can also decrease due to the direct access to the user. Figure 2 shows the flowchart of the request generation and handling using the proposed IoT architecture. Users and devices generate requests in the application and session layers. The application and edge-side layers will check whether it is necessary to send these requests to the cloud server or not. After examining the type of request, if the request can be handled locally, then the network conditions will be checked. If the deployed network has some local networking with LAN WiFi, Bluetooth, or Zigbee, then the request will be checked and forwarded to the deployed network. Such requests can be handled locally, and there is no need to send them to the IoT platform server. If the request is not local, then the IoT platform has to be accessed. The same procedure must be followed if the network does not exhibit any local networking.

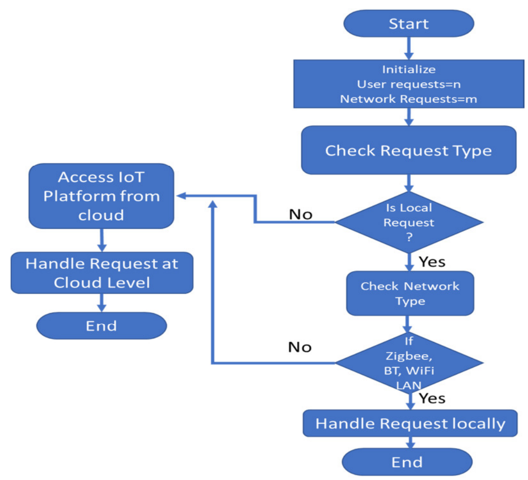

Fig. 2. Flowchart of the proposed IoT framework.

V. SIMULATION RESULTS

A simulation model was created to examine the proposed architecture. The numbers of requests generated by the user-side layer and the edge-side nodes were combined for the IoT service platform, and it was assumed that the number of service requests generated per node per unit of time was between 1 and 10, 30, and 50. In this scenario, the number of IoT nodes increases as time increases, and the load over the number of requests generated per unit of time can be formulated as:

$$R = \sum_{i=1}^{n} rand(R(i)) \quad (1)$$

where $R(i)$ is the number of requests generated by the i-node at any time t. Figure 3 shows the number of requests generated at any time t in a stochastic nature. The graph also shows that the number of requests continuously increases for an increasing number of IoT nodes. The number of requests was considered stochastic due to their random nature. As Figure 4 shows, applying the proposed method decreased the total number of requests. This was due to bypassing some user requests directly to the edge side layer. Figures 3 and 4 show that requests may proceed to 4500 requests per time unit using the conventional framework and a maximum of 2600 requests per time unit using the proposed method.

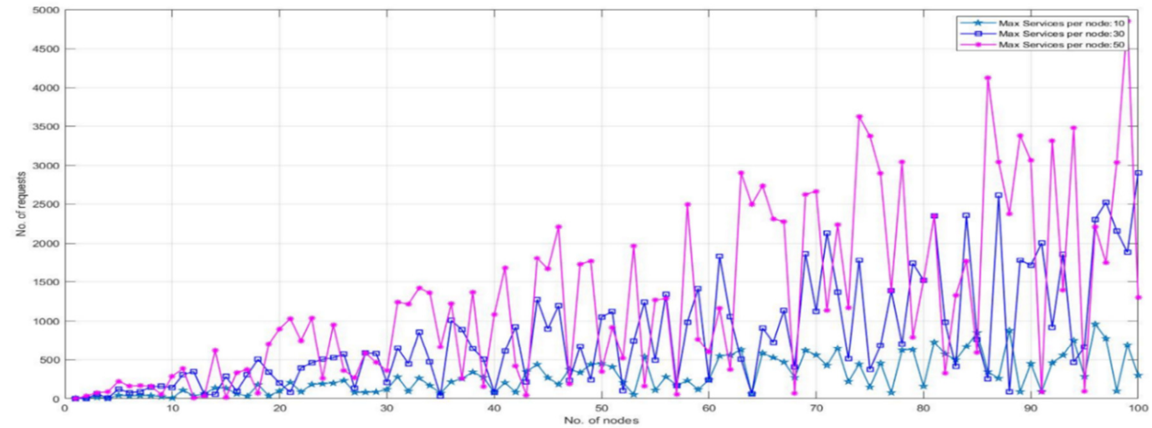

Fig. 3. Request generation by a conventional network of 100 nodes.

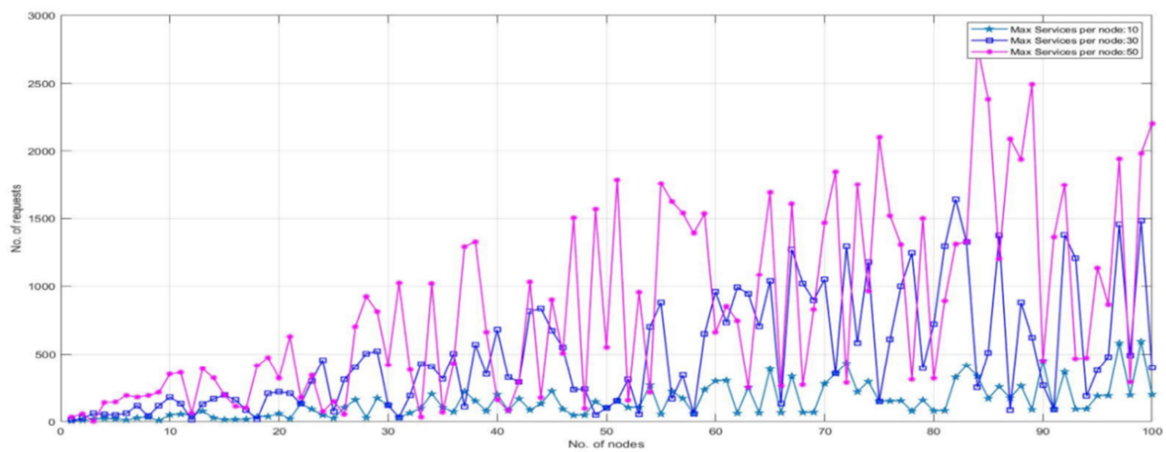

Fig. 4. Request generation by the proposed network of 100 nodes.

A test environment was built in Matlab to validate and justify the proposed method, considering a Wi-Fi network. This network requires a Wi-Fi Access Point (AP) and a Station Point (STA). A 12-node AP and a 24-node STA model were considered and used for the development of a Wi-Fi-enabled IoT model for a BMS application. Some network parameters were considered to examine the load sharing of the server. The IoT servers were accessed by the AP, which was responsible for sending and receiving data from the STAs to the server and vice versa. Figure 5 shows the throughput of 12 APs in the proposed and conventional approaches. The number of requests generated by the STAs was taken in two domains. In the first domain, the requests were forwarded to APs directly without using any filtration process, and in the second, the requests were forwarded to APs with the intelligent queuing process. This process followed the algorithm shown in Figure 3 to separate the server-oriented and local requests. Figure 5 shows that the throughput of the AP improved significantly in each node, increasing the quality of the network. Figures 6 and 7 show the packet loss and the average packet latency of STAs for both methods, considering again 24 STAs.

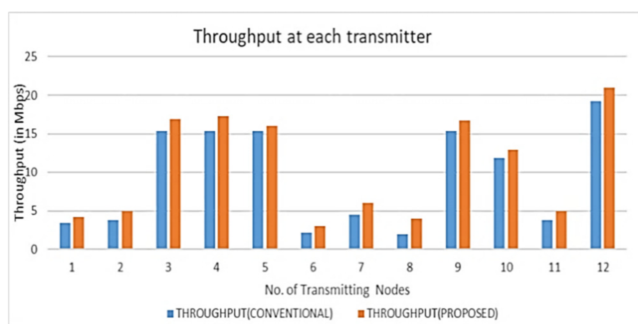

Fig. 5. Throughput of conventional and proposed scheme.

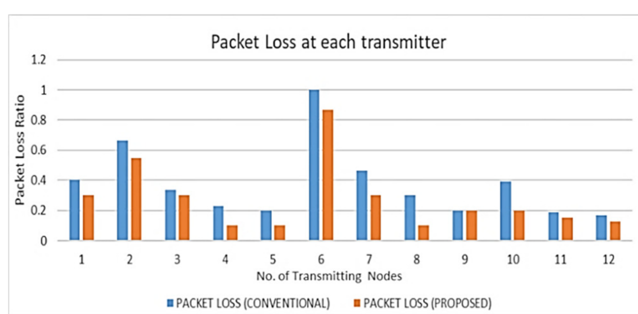

Fig. 6. Packet loss for conventional and proposed frameworks.

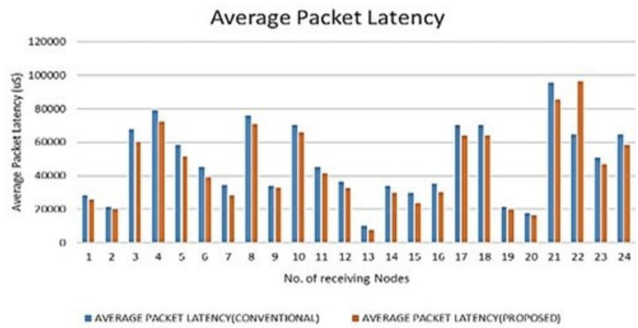

Fig. 7. Average packet latency for proposed and conventional schemes.

Figure 5 shows the reduction in packet loss for each node that functions as an STA. STAs are nodes that have some sensors, and may connect directly with the user based on the request type. Conventionally, each STA is supposed to provide and receive data from the server, and any request generated by the user or the STA pings the server and increases its load and packet latency. The proposed method reduced the number of requests to the server, and thus reduced the packet loss on each node. Figure 7 shows the average packet latency for the proposed and the conventional method. Average packet latency also decreased using the proposed algorithm, due to the smaller number of requests to the server. Packet latency is directly proportional to the number of requests generated by each STA. The number of requests generated for any IoT platform using the conventional approach increases significantly with the number of nodes, lowering the QoS parameters of the IoT network. As the increasing number of nodes in an IoT platform can't be controlled, the proposed approach can suit many applications. The proposed algorithm helps to lower the requests without limiting the number of nodes, and hence it can be very useful. Future work could investigate an AI-based intelligent approach to control the number of requests to the server at the device level. The nature of such a modification should have a very small memory footprint because of limited resources.

VI. CONCLUSION

Enhancing QoS without compromising the number of nodes is a challenge for IoT applications and the research society. The most important QoS parameters of an IoT network, i.e. packet loss, throughput, and latency, are directly proportional to the number of packets generated by the network. An intelligent model was proposed and investigated to reduce the requests to the server from the nodes. This model recognized requests that can be handled locally or should be forwarded to the server. The packet loss, throughput, and packet latency of the proposed model were examined using Matlab. The results showed that the proposed method minimized the number of requests by almost half on average of the conventional model. This enhanced the throughput of the networks by around 5- 10%, depending on the number of nodes, and decreased the average packet latency by 4%. The average packet loss also decreased by around 3-6%.

Dividing the number of requests to those to be handled locally and by the server can become smarter using AI and machine learning algorithms, which could be future work. AI and machine learning algorithms can drastically enhance the proposed model and enhance the overall network performance. A trained neural network and a double Q method could enhance QoS. For validation, network tools like NS2, NS3, or a real-time physical network could be utilized.

REFERENCES

[1] L. Li, S. Li, and S. Zhao, "QoS-Aware Scheduling of Services-Oriented Internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1497–1505, Feb. 2014, https://doi.org/10.1109/TII.2014.2306782.

[2] B. Guo, D. Zhang, Z. Wang, Z. Yu, and X. Zhou, "Opportunistic IoT: Exploring the harmonious interaction between human and the internet of things," *Journal of Network and Computer Applications*, vol. 36, no. 6, pp. 1531–1539, Nov. 2013, https://doi.org/10.1016/j.jnca.2012.12.028.

[3] J.-S. Leu, C.-F. Chen, and K.-C. Hsu, "Improving Heterogeneous SOA-Based IoT Message Stability by Shortest Processing Time Scheduling," *IEEE Transactions on Services Computing*, vol. 7, no. 4, pp. 575–585, Jul. 2014, https://doi.org/10.1109/TSC.2013.30.

[4] Y. Ding, Y. Jin, L. Ren, and K. Hao, "An Intelligent Self-Organization Scheme for the Internet of Things," *IEEE Computational Intelligence Magazine*, vol. 8, no. 3, pp. 41–53, Dec. 2013, https://doi.org/10.1109/MCI.2013.2264251.

[5] P. Vlacheas *et al.*, "Enabling smart cities through a cognitive management framework for the internet of things," *IEEE Communications Magazine*, vol. 51, no. 6, pp. 102–111, Jun. 2013, https://doi.org/10.1109/MCOM.2013.6525602.

[6] D. Minoli, K. Sohraby, and B. Occhiogrosso, "IoT Considerations, Requirements, and Architectures for Smart Buildings—Energy Optimization and Next-Generation Building Management Systems," *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 269–283, Oct. 2017, https://doi.org/10.1109/JIOT.2017.2647881.

[7] N. Sharghivand, F. Derakhshan, L. Mashayekhy, and L. Mohammad Khanli, "An Edge Computing Matching Framework with Guaranteed Quality of Service," *IEEE Transactions on Cloud Computing*, pp. 1–1, 2020, https://doi.org/10.1109/TCC.2020.3005539.

[8] K. Cao, G. Xu, J. Zhou, T. Wei, M. Chen, and S. Hu, "QoS-Adaptive Approximate Real-Time Computation for Mobility-Aware IoT Lifetime Optimization," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 38, no. 10, pp. 1799–1810, Jul. 2019, https://doi.org/10.1109/TCAD.2018.2873239.

[9] Y. Hu, Y. Li, M. C. Gursoy, S. Velipasalar, and A. Schmeink, "Throughput Analysis of Low-Latency IoT Systems With QoS Constraints and Finite Blocklength Codes," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 3093–3104, Mar. 2020, https://doi.org/10.1109/TVT.2020.2968463.

[10] A. Čolaković and M. Hadžialić, "Internet of Things (IoT): A review of enabling technologies, challenges, and open research issues," *Computer Networks*, vol. 144, pp. 17–39, Oct. 2018, https://doi.org/10.1016/j.comnet.2018.07.017.

[11] M. Aazam and K. A. Harras, "Mapping QoE with Resource Estimation in IoT," in *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, Limerick, Ireland, Apr. 2019, pp. 464–467, https://doi.org/10.1109/WF-IoT.2019.8767254.

[12] A. Khamosh, M. A. Anwer, N. Nasrat, J. Hamdard, G. S. Gawhari, and A. R. Ahmadi, "Impact of Network QoS factors on QoE of IoT Services," in *2020 - 5th International Conference on Information Technology (InCIT)*, Chonburi, Thailand, Jul. 2020, pp. 61–65, https://doi.org/10.1109/InCIT50588.2020.9310969.

[13] S. Javed, S. Ghazala, and U. Faseeha, "Perspectives of Heat Stroke Shield: An IoT based Solution for the Detection and Preliminary Treatment of Heat Stroke," *Engineering, Technology & Applied Science Research*, vol. 10, no. 2, pp. 5576–5580, Apr. 2020, https://doi.org/10.48084/etasr.3274.

[14] Y. Djeldjeli and M. Zoubir, "CP-SDN: A New Approach for the Control Operation of 5G Mobile Networks to Improve QoS," *Engineering,*

Technology & Applied Science Research, vol. 11, no. 2, pp. 6857–6863, Apr. 2021, https://doi.org/10.48084/etasr.4016.

[15] N. K. Al-Shammari, T. H. Syed, and M. B. Syed, "An Edge – IoT Framework and Prototype based on Blockchain for Smart Healthcare Applications," *Engineering, Technology & Applied Science Research*, vol. 11, no. 4, pp. 7326–7331, Aug. 2021, https://doi.org/10.48084/etasr.4245.

[16] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010, https://doi.org/10.1016/j.comnet.2010.05.010.

[17] H. Yin, Z. Wang, and N. K. Jha, "A Hierarchical Inference Model for Internet-of-Things," *IEEE Transactions on Multi-Scale Computing Systems*, vol. 4, no. 3, pp. 260–271, Jul. 2018, https://doi.org/10.1109/TMSCS.2018.2821154.

[18] Ian Beavers, "Intelligence at the Edge Part 1: The Edge Node," Northwood, MA, USA:Analog Devices Inc., 2017.

[19] S. Al-Sarawi, M. Anbar, K. Alieyan, and M. Alzubaidi, "Internet of Things (IoT) communication protocols: Review," in *2017 8th International Conference on Information Technology (ICIT)*, Amman, Jordan, Feb. 2017, pp. 685–690, https://doi.org/10.1109/ICITECH.2017.8079928.

[20] S. N. Shirazi, A. Gouglidis, A. Farshad, and D. Hutchison, "The Extended Cloud: Review and Analysis of Mobile Edge Computing and Fog From a Security and Resilience Perspective," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 11, pp. 2586–2595, Aug. 2017, https://doi.org/10.1109/JSAC.2017.2760478.

[21] T. Vu, C. J. Mediran, and Y. Peng, "Measurement and Observation of Cross-Provider Cross-Region Latency for Cloud-Based IoT Systems," in *2019 IEEE World Congress on Services (SERVICES)*, Milan, Italy, Jul. 2019, vol. 2642–939X, pp. 364–365, https://doi.org/10.1109/SERVICES.2019.00105.

[22] A. Azari, Č. Stefanović, P. Popovski, and C. Cavdar, "On the Latency-Energy Performance of NB-IoT Systems in Providing Wide-Area IoT Connectivity," *IEEE Transactions on Green Communications and Networking*, vol. 4, no. 1, pp. 57–68, Mar. 2020, https://doi.org/10.1109/TGCN.2019.2948591.

[23] M. Shahzad and A. Ganji, "IoTm: A Lightweight Framework for Fine-Grained Measurements of IoT Performance Metrics," in *2018 IEEE 26th International Conference on Network Protocols (ICNP)*, Cambridge, UK, Sep. 2018, pp. 12–22, https://doi.org/10.1109/ICNP.2018.00012.

[24] G. Tanganelli, C. Vallati, and E. Mingozzi, "CoAPthon: Easy development of CoAP-based IoT applications with Python," in *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, Milan, Italy, Sep. 2015, pp. 63–68, https://doi.org/10.1109/WF-IoT.2015.7389028.

[25] S. Y. Jeon, J. H. Ahn, and T.-J. Lee, "Data Distribution in IoT Networks with Estimation of Packet Error Rate," in *2016 10th International Conference on Next Generation Mobile Applications, Security and Technologies (NGMAST)*, Cardiff, UK, Dec. 2016, pp. 94–98, https://doi.org/10.1109/NGMAST.2016.25.

[26] F. H. Hung *et al.*, "Packet error rate analysis in IoT for industrial air conditioning system," in *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, Beijing, China, Jul. 2017, pp. 8367–8370, https://doi.org/10.1109/IECON.2017.8217469.

[27] N. Chhabra, "Comparative Analysis of Different Wireless Technologies," *International Journal of Scientific Research in Network Security and Communication*, vol. 1, no. 5, pp. 13–17, 2013.

[28] J. Zhou, X. Gong, L. Sun, Y. Xie, and X. Yan, "Adaptive Routing Strategy Based on Improved Double Q-Learning for Satellite Internet of Things," *Security and Communication Networks*, vol. 2021, Apr. 2021, Art. no. e5530023, https://doi.org/10.1155/2021/5530023.