International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Virtual Edge Computing Architecture Model for the Real-Time Data Server in the IoT Environment

Kuldeep Sharma[1], Arun Malik[2]

Submitted: 24/08/2022 Accepted : 21/11/2022

Abstract:Recently, a number of smart devices are connected through the Internet to achieve data processing and generation. The data generated from the cloud server demand robust and reliable data storage and protection for unauthorized data access. Additionally, the processed data demands for avast range of processing power to tangible effective information for processing. The different business process comprises of technologies to increase efficiency and performance with the reduced cost of operation in the IoT devices. The data processing leads to the data handledwith the edge computing technology. The technological procession deal with the response time improved cost-saving bandwidth and battery life those significantly impacts on safety and privacy in the organization. This paper presented a Virtual Environment Based HTTP server model termed as (VEnvMQTT) for effective data processing in the edge computing architecture of the IoT environment. The proposed VEnvMQTT model comprises of the actuators and IoT server data. The proposed VEnvMQTT model evaluates the data collected from the sensor at an effective level of processing time with edge computing technology. The analysis of the simulation results expressed thatthe proposed VEnvMQTTmodel achieves a reduced latency of 24.566ms which is ~20% minimal to the existing MQTT model.

Keywords:*Virtual server, Edge computing. MQTT, Internet of Things (IoT), Quality of Sevices*

1. Introduction

Internet of Things is the system of physical gadgets such as vehicles, home appliances, and different devices connected to the Internet. These devices are equipped with software, hardware, network, sensors, and actuators, which allow these “things” to connect and exchange data [1]. This interconnection opens the door for more straightforward coordination of the real world ‘things’ into computer-based frameworks. This binding improves proficiency changes, commercial advantages, and reduced human efforts. A survey conducted in 2017 states that there were 8.4 billion IoT gadgets during that year, and it also appraises that “there will be around 30 billion gadgets by 2020” [2]. The worldwide market estimation of “IoT is anticipated to be $7.1 trillion by 2020” [3]. "IoT" was coined by Kevin Ashton in 1999. However, the author favors the expression called "Web of Things" (WoT) [4]. At that point, “Radio Frequency Identification” (RFID) as a primary part in building IoT would enable computers to deal individual things.

With the explosion of information, devices, and networking, the present Cloud architecture on its own cannot handle the flood of data. The Cloud offers extensive computation, storage, and even connectivity for users who can access the Cloud efficiently and economically. These centralized resources can create notable delays and lack in the performance for devices located far away from a centralized Cloud viz., data center. Edge computing, otherwise called merely "Edge," does processing near the data source, and should not transmit to the remote Cloud or other unified frameworks for handling. The time it takes to send data to the source is reduced, which enhances the speed and execution of data transmission [5]. Fog computing is a standard that characterizes how edge processing should function. It also encourages the task of figuring, stockpiling, and system administration benefits between end-to-end devices, where data values are identified in the Cloud. Furthermore, Fog off-loading the Cloud for edge processing [6] can resort to many applications. There is a range of critical abilities which Edge can provide for Industrial IoT (IIoT) applications taking into account the prerequisites of the current issues, as enunciated by the accompanying instances found in regular mechanical tasks [7]. With Edge, PC, and capacity frameworks dwell at the edge also, as close as conceivable to the segment, gadget, and application or human that creates the data being handled. The objective is to minimize latency because the data need not be sent from the edge of the system to a central organizing framework, and back to the edge [8]. The IoT-associated gadget is an obvious use for edge processing. With remote sensors introduced on a machine, partly or entirely, they produce large measures of data. The data that is sent over an extended network is to be dissected, logged, and followed. This takes substantially more time than if the information is handled at the edge, near the data [9].

IoT demands a fresh kind of infrastructure which requires more capital. Present Cloud computing models are not planned for the variety, volume, and velocity of information which these IoT devices produce. For example, current unconnected devices are present in the volume of billions, and they are producing more

[1]*Department of Computer Science and Engineering, Lovely Professional University, Phagwara, Jalandhar, Punjab, India*
ask.kuldeep@gmail.com
[2]*Department of Computer Science and Engineering, Lovely Professional University, Phagwara, Jalandhar, Punjab, India*
arun.17442@lpu.co.in

than two Exabytes, that is 2 billion GB of data each day [10]. By 13 the year 2020, it is expected that around 50 billion devices will be connected to the existing network. Moving all the data from the things to the Cloud for processing would require a massive amount of transmission capacity. Some are machines that associate with controllers utilizing modern protocols, not Internet Protocol (IP). Before data can be transmitted to the Cloud for processing or storage, it is predesignated with IP, and these changes increase the transmission delays. In a few ventures, milliseconds matter very much, when the anticipated response is delayed, leading to production line shut down. For instance, seaward oilrigs create 500 GB of data weekly [11], and commercial planes produce 10 TB of data at regular intervals of flight travel. For such situations, appropriated confined handling is required as close with regards to the wellspring of the data age. There are many challenges seen in IoT framework about network latency, processing delay, and bandwidth over the critical measure of data being created by the end devices [12]. Fog computing stretches out computing and administration to the edge of the system, like Cloud. Fog offers data processing, stockpiling, and application administrations to end-clients. For sensors, routing the information safely to Fog where all analytics are being performed, no proper routing algorithm exists in place. Most of the routing algorithms in IoT are targeted at wireless sensor levels and up to sensor gateway or cluster head and not for IoT[13]. Sensors and other objects in IoT communicate to Cloud, where a lot of IoT protocols and communication methods are available [14]. IoT protocols did not take into consideration Fog computing as IoT was conceived with Cloud as a source for performing analytics. IoT involves heterogeneous types of data like health, agriculture, energy, etc. So, it is challenging to come with one standard routing algorithm or protocol that has been developed for existing IoT models [15]. Handling such heterogeneous data from different sensors requires a minimal level of analytics for predicting the data pattern and autonomously coordinating among the Fog routers for fast data processing and transmission across the objects. In this paper presented a Virtual environment based MQTT model for the IoT data processing. The experimental analysis presented thatthe proposedVEnvMQTT model achieves the reduced latency compared to the conventional MQTT.

2. Related Works

"Routing Protocol for Low-Power and Lossy Networks" (6LoWPAN) face tremendous challenging issues in routing viz., Resource constraints, frequent topology changes, and multi-hop networking. Specific application requirement should be considered for both IPv6 and 6LoWPAN components [16]. The Routing over Low power Lossy Network Working Group [ROLL WG] of IETF proposed the [ROLL WG] the Routing over Low power Lossy Network called the RPL. A good amount of support is provided by RPL protocol for connection layers that are constrained, lossy, etc. RPL is predominantly used in host or switch gadgets with significant benefit in building/home automation and urban applications [17]. It can rapidly develop and organize routes, appropriately learn routes among hubs, and adjust the topology in an extremely effective manner. In the most regular setting of RPL, the hubs of the system are connected to the root node or gadgets which are routed in a multi-hop manner. Also, the root node is responsible for data accumulation and coordination [18].

Bluetooth is essentially a short-range exchange innovation which has been vital in various customer-related products. It is very much essential in wearable devices, which are very much associated with IoT [3]. Initially, Bluetooth was available in the Cell phone to start. "Bluetooth Low Energy (BLE)" either "Bluetooth Smart " is one of noteworthy protocol in the field of IoT. BLE has been developed for reduced power utilization [4]. ZigBee Remote Control (RF4CE), and ZigBee PRO are the accessible Zigbee profiles among other accessible ZigBee profiles [5]. According to the IEEE 802.15.4 protocol, Zigbee is working at 2.4 GHz, specifically for the applications that demands a moderately inconsistent information flow at less data rate over a 100-meter range [6]. Zigbee/RF4CE has strong focal points in complex systems with high security, easy accessibility, low power, great versatility with more nodes. The features mentioned above make the utilization of this protocol more in the wireless sensor network in M2M and IoT applications [7].

Sigfox mainly focusses on separating Wi-Fi and Cellular communication. It utilizes the ISM groups, which are allowed to communicate without the requirements of any security licenses. The motivation behind Sigfox is that some M2M applications keep on running over a small power backup and also demands a lower level of information exchange. As "WiFi's" coverage is squat, and mobile communication has been excessively costly, Sigfox is advantageous for m2m application [8]. Sigfox utilizes an innovation called "Ultra Narrow Band" (UNB) and performs low data exchange at the rate of ten to thousands of bits per second. Sigfox expands just "50 microwatts" contrasted with "5000 microwatts" for mobile information exchange or can convey a regular standby period which is almost "20 years with a 2.5Ah battery," but it is just a few months, i.e., 0.2 years for the cell. The system is right now being taken off in urban areas crosswise over Europe, incorporating ten urban areas in the United Kingdom, for instance [9]. System compromises a strong, control effective and versatile system that can communicate with a considerable number of battery-worked gadgets crosswise over territories of a few square kilometers, making it appropriate for different M2M applications that rely upon incorporating in Smart meters, quiet screens, security gadgets, road lighting, and natural sensors [10]. The Sigfox framework utilizes chipset, for example, the EZRadioPro remote handsets from Silicon Labs, which conveys industry-driving remote execution, broadened run and ultralow power utilization for remote systems administration frameworks working in the sub 1GHz band [11].

3. Virtual Edge MQTT model for QoS in IoT

The word edge computing approach denotes a decentralized system and an extension of Cloud computing with limited control and administration. The proposed VEnvMQTT model computing permits the Cloud-based administration to extend its range to edge of a system to offer nearby and active openness to edge gadgets. Generally, edge computing system has two planes, these planes are given below:

Data plane: it is referred to as forwarding and control planes. This plane figures out the result of the information packets. It enables registering assets to be put at any place in the system as they do not need to be focused on a server as they can be appropriated on the edge of the system.

Control plane: It gives an sketch of the system and its capacities with directing conventions that keep running in the building control component. Fog enables IoT information to be prepared in an information Centre point or smart gadget nearer to the sensor that is producing it [19]. The classification for the edge computing is presented in figure 1.

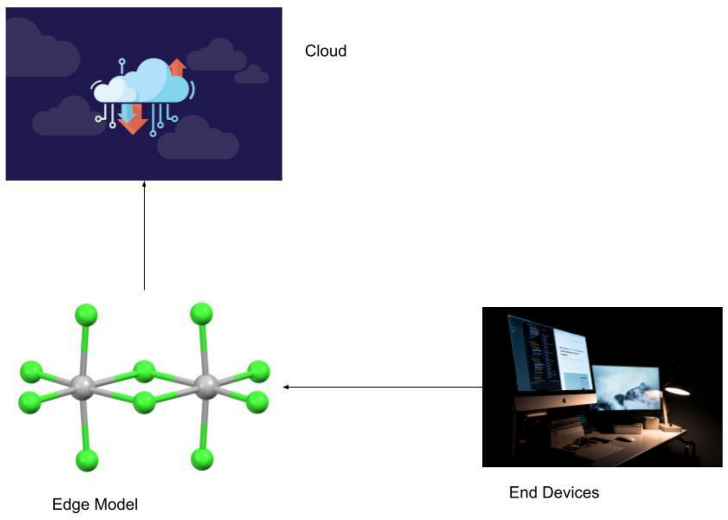

Figure 1: Fog Computing Model

In the proposed VEnvMQTT Cloud, it is dependably needed to rely upon the Cloud storage and getting the required data transfer speed and availability. On account of Fog computing, the data can be brought to the gadgets locally for processing without having to process in the Cloud. This processing will support openness, convenience, and relative ease of use of gadget data. The rise of Fog computing will support coordinated effort among gadgets and data centers. The world is seeing an enormous flood concerning digitally produced data from smart things and connected gadgets. Then again, with the ascent of smartphones, applications empowering, and ever increasing number of clients to get to data, CPU power, control, and communication with their end gadgets are increasing continuously.

The implementation ofVEnvMQTTfor Fog/Edge computing with Main and Remote brokers has been carried out in Python based Virtual Fog Simulator called PVFOG Simulator. The figure 2 shows the control flow in VEnvMQTT Simulator with Remote and Main Broker based on proposed Publish/Subscribe Model. Now before going into the details of VEnvMQTT Simulator, Discrete Event Simulation is discussed below.

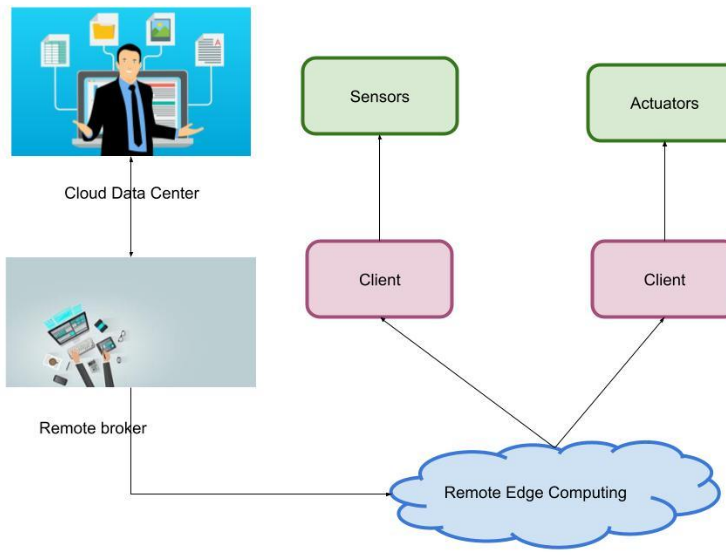

Figure 2: Process Flow in VEnvMQTT

The steps involved in the working of the proposed system is as follows:

Step 1: Multiple virtual sensors are created, and these sensors are grouped into a single class called IoT device that represents the virtual device to imitate the real world systems like Arduino and other microcontroller boards.

Step 2: All the sensor instances are running in parallel mode, and these sensors are using the discrete event simulation methods derived from the Simpy.

Step 3: Then the Virtual IoT devices will have the running instant of the MQTT Client.

Step 4: These clients are bound with the Remote Broker using the Remote Broker's IP address and specified a port.

Step 5: All virtual devices are running in the parallel mode so that sensors data will be posted to the virtual IoT devices at different time intervals.

Step 6: This virtual IoT device will post these sensor data to the Remote Broker using the respective clients.

Step 7: The Remote Broker will receive the data and do the processing at the Fog level

Step 8: If any value lies below the threshold value, then the remote Fog router will resend the data to the main broker.

Step 9: Else the Remote Broker will send the message to the subscribed clients directly

The proposed VEnvMQTTcomputational approach assigns the responsibility for each layer as well as each tier. This proposed method contains a few assumptions and, a mathematical model has been derived as follows.

• The nodes represented as end nodes here are the nodes that contain any one of the following: PIC, Arduino, Intel Galileo, Raspberry Pi, and other microprocessor-based devices.

• Each end nodes can transfer the sensed information via the network to any other end nodes and/or Fog/Edge nodes.

• MQTT clients exist for all kinds of embedded boards like Arduino and Intel with different programming language options. These MQTT client libraries can be deployed in these boards easily.

• Nodes assigned as Fog nodes have the capability to exchange information amongst each other by sharing the MQTT 'Publish/Subscribe' method topics.

• One MQTT broker is deployed in the Fog and another one in the Cloud level, these two brokers are implemented using Python language. The MQTT protocol over TCP/IP has been used to establish communication among the brokers via the backhaul network. The elements used in this mathematical model to calculate different latencies are discussed below.

3.1 IoT Devices and Sensors/Actuators with Virtual Environment MQTT

Modern embedded devices, such as Arduino, Raspberry Pi, etc., have enough computational resources as well as network communication in both wireless and wired environments. Symbol δ denotes the end device. δ has various components, like sensors and actuators. Seven tuples are there in the δ which are given in equation (1).

$$\delta = < D_{id}, D_{st}, \sigma[x], \alpha[y], L, H, c[z] > \quad (1)$$

A unique integer value has been represented by D_{id}in theδ. The status of each δ has been denoted by a binary flag called D_{st} . The symbol σ is used to represent the sensors and σ is denoted by the one-dimensional array. The four tuples used in the $\sigma[x]$ is presented in equation (2)

$$\sigma = \langle \sigma_{id}, \sigma_{st}, \sigma_{ip}, A_{id} \rangle \quad (2)$$

σ_{id}represents the unique id of the σ and status of the same is denoted by σ_{st}. i={i1,i2,..,ip}., this mathematical set denotes a specific node which is currently being sensed where the information is represented by the notation σ_{id}. There is a possibility that multiple a number of MQTT clients can be associated with the single σ_{ip}. So, this MQTT clients are represented by theA_{id}. This also represents the software entity which contains the application instance details and these, clients transmit the information to the VEnvMQTT brokers. The α denotes the actuators, which are represented using a one-dimensional array that contains y elements. There is four tuples which are given in equation (3).

$$\alpha = \langle \alpha_{id}, \alpha_{st}, \alpha_{jp}, A_{id} \rangle \quad (3)$$

where,

α_{id}- unique id of a particular actuator α_{st},- Binary flag which symbolizes the status of α_{id}. α_{jp}- Event of the actuator, where j={j1,j2,..,jp} and A_{id}. The Software entity and the list of VEnvMQTT clients associated with α_{id}.

The clients who are subscribed to the topic in the VEnvMQTTbroker will receive the information. Whenever a new message is published by another client, it will be notified to this client via the VEnvMQTT broker. Generally, the IoT is spread across a wide geographical area to denote the location of a particular δ. Location information is a primary one whenever the important decisions are taken up in an automated system. l is represented by four tuples and the same are given in equation (4).

$$l = \langle l_x, l_y, l_z, ts.H \rangle \quad (4)$$

where, lx - longitude ly - latitude lz- altitudes ts – time stamp and H – Hardware configuration

The hardware configuration contains four items that are represented by four tuples, as shown in equation (5)

$$H = \langle h_{cpu}, h_{mem}, h_{bat}, com[x] \rangle \quad (5)$$

where, h_{cpu}- Processing capacity of δ h_{mem} Storage capacity of δ, which includes both RAM and ROM h_{bat}- Represents the status, capacity, and remaining power of the battery and com[x] – the one-dimensional array to denote the status of communication methods Nowadays, multiple wireless communication methods exist in the same IoT device, which is denoted by a one-dimensional array to represent the three items are computed using the equation (6).

$$x_{com} = x_{ctype}, x_{cfreq}, x_{size} \quad (6)$$

where, x_{ctype}- Communication type x_{cfreq} –Frequency of a particular communication method Coverage range of x_{size}. In equation (7) represents the list of VEnvMQTT clients deployed in the δ, which is denoted by c[z], which is one-dimensional array of three elements.

$$c = \langle c_{id}, c_{st}, c_{host} \rangle \quad (7)$$

where, c_{id}– a unique identification number of the MQTT client, c_{st}- Binary flag to represent the status of the particular MQTT client and c_{host}- Host address of the MQTT broker.

Edge nodes are denoted by the symbol φ, and each of these φ has the options for running the multiple MQTT brokers. These brokers are called as Remote Broker (RB). RB is connected with MQTT clients through different kinds of communication techniques. RB is denoted by the symbol β_{id}. This is mathematically represented in equation (8).

$$\beta_{rb} = < \beta_{id}, c[v], f_{id} > \quad (8)$$

where, β_{id} - Unique identification number of the β_{rb} , f_{id}- Unique identification number of the φ, $c[v]$- List of connected VEnvMQTT clients represented as c[v] is denoted by the one-dimensional array, and this count of connected MQTT clients are denoted by v. In this scenario, multiple β_{rb} can be deployed in the same φ.The multiple β_{rb} can be deployed in the same φ, because these $\beta_{rb}$are software entities which are directly connected with the main broker which is denoted by the β_{cb}. The many-to-many relationship has been used between the δ and β_{rb}. In equation (9) represents the mapping of the $\beta_{rb}$to their respective φ

$$f(,) = \widetilde{\beta_{rbi} \rightarrow \widetilde{\varphi}_i} \quad (9)$$

The equation (10) represents the mathematical form of φ, which contains three tuples.

$$\varphi = < F_{id}, F_{st}, C_{ap}, N[x], \beta_{rb}[y] > \quad (10)$$

F_{id}- Unique identification number of φ ,F_{st}- Current status of φ,C_{ap}- Access point to the Cloud environment, $N[x]$ - Shortest route information to adjacent φ nodes and,$\beta_{rb}[y]$ – List of β_{rb} instances in one-da imensional array with y elements. In equation (11) represents the mapping between the $\beta_{rb}$and β_{cb}. Initially the γ, will be mapped to the $\beta_{rb}$and will be mapped to the β_{cb}.

$$f'(,) = \widetilde{\beta_{rb} \rightarrow \beta_{cb}} \quad (11)$$

Once all these mapping procedures between different devices are completed, route information amongst different β_{rb} will be discovered and stored for further usage.

4. Simulation Analysis

The different kinds of latencies are used to measure the performances of the VEnvMQTT based traditional IoT model and the proposed multi-tier Fog computational model. The mathematical form of these matrices. In table 1 presented the measured simulation parameters considered are presened.

Table 1: SimulationEnvironment

VEnvMQTT based IoT		**Multi-Tier Architecture**	
Virtual Sensor	100	Virtual Sensor	100
Virtual Actuators	100	Virtual Actuators	100
IoT devices	100	IoT devices	100
Remote Brokers	5	Remote Brokers	10
Roof Main broker	3	Roof Main broker	1
Simulation time	2000	Simulation time	2000

The table 2 provides the comparative analysis of the packet transmission count for the varying number of clients.

Table 2: Comparison of Packet Count

Number of Clients	**Traditional MQTT**	**Proposed VEnvMQTT**
1000	1767	1456
2500	6348	5987
5000	12398	10934
7500	14936	11387
10000	18369	16298

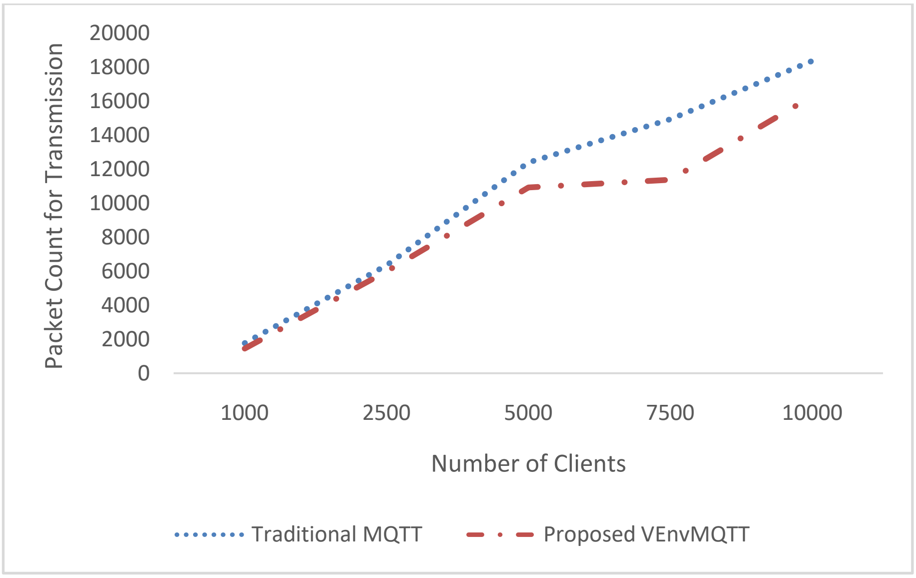

Figure 3: Comparison of Packet Count

The performance of the proposed service latencies for the varying client are computed. The examination expressed that proposed VEnvMQTT model achieves the minimal service latency compared with the traditional MQTT model. The figure 3 provides the comparative analysis of the packet count. The table 2 provides the comparative analysis of the proposed VEnvMQTT with the conventional MQTT model is presented.

Table 2: Comparison of Service Latency

Number of Clients	**Traditional MQTT (ms)**	**Proposed VEnvMQTT (ms)**
1000	13.67	8.73
2500	17.45	9.83
5000	19.83	11.84
7500	21.85	13.67
10000	28.74	16.83

Table 3: Comparison of Transmission latency

Number of Clients	**Traditional MQTT (ms)**	**Proposed VEnvMQTT (ms)**
1000	9.84	4.34
2500	18.94	7.83
5000	23.72	10.83
7500	29.74	12.85
10000	33.64	14.78

Table 4: Comparison of Processing Latency

Number of Clients	**Traditional MQTT (ms)**	**Proposed VEnvMQTT (ms)**
1000	4.667	2.876
2500	4.893	2.969
5000	5.672	3.045
7500	6.782	3.567
10000	7.346	3.856

In table 3 and table 4 provides the transmission and processing latency for the varying number of clients ranges from 1000 to 1000. The comparative analysis expressed that proposed VEnvMQTT model exhibits reduced latency than the traditional MQTT. Figure4 – 6 provides the comparative analysis of the proposed VEnvMQTT with the traditional MQTT is presented.

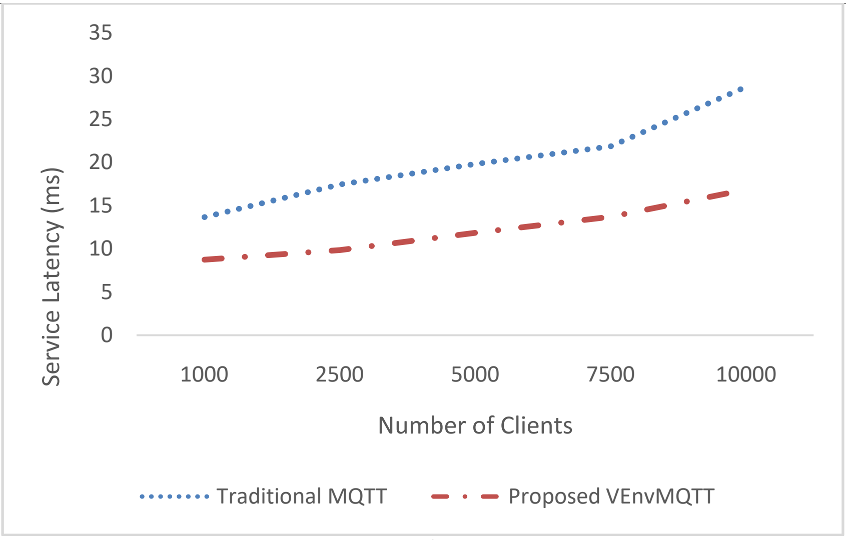

Figure 4: Comparison of Transmission Latency

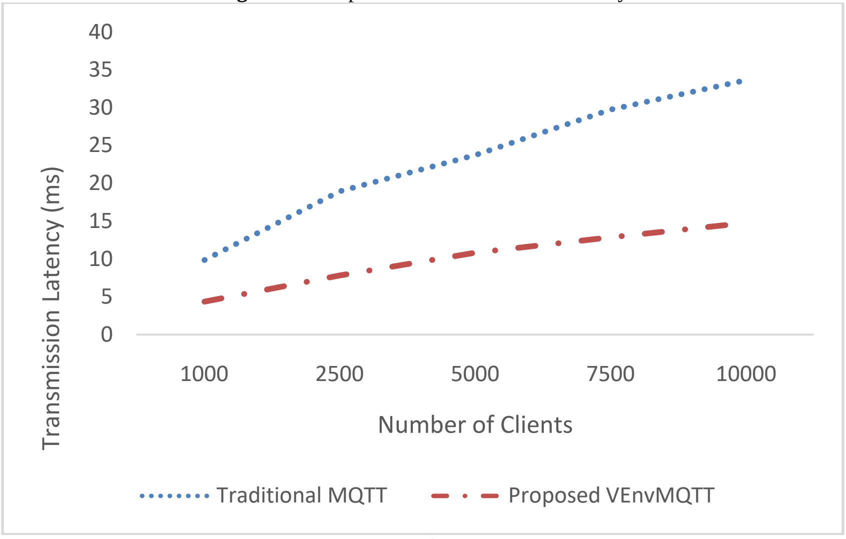

Figure 5: Comparison of Transmission Latency

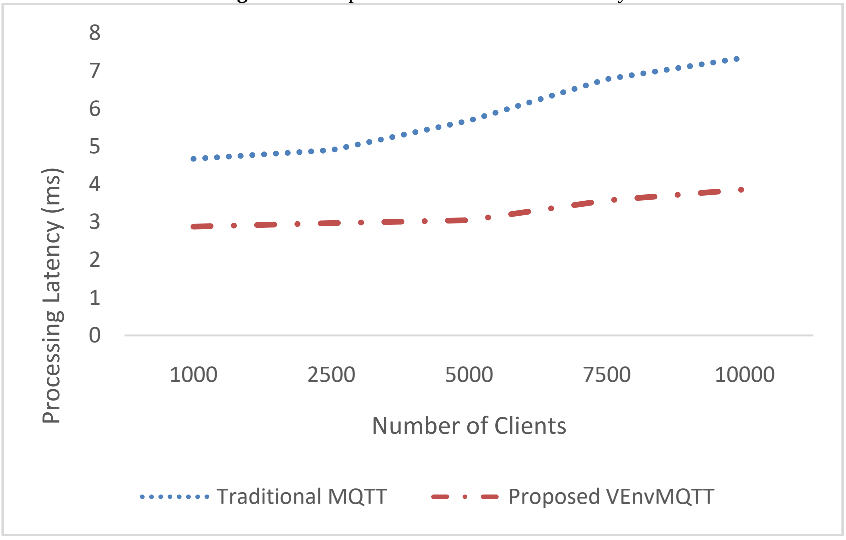

Figure 6: Comparison of Processing Latency

The simulation analysis expressed that processing increases with the queue probability of waiting time in the edge computing environment. The processing is achieved higher rate of 20% for the proposed VEnvMQTT model higher than the traditional MQTT model. With the edge computing model the performance is achieved with the cloud analytics for the IoT platform. The transmission latency is 35% minimal than the conventional MQTT. Through analysis it is concluded that proposed VEnvMQTT model achieves the higher QoS compared with the traditional MQTT model.

5. Conclusion

To increases the overall data processing with the edge computing environment IoT data is evaluated in the virtual environment. The proposed VEnVMQTT model performance is evaluated compared with the traditional MQTT model. The simulation is performed for the varying latency count such as service latency, transmission latency and processing latency. The simulation examination expressed that proposed VEnvMQTT model exhibits the reduced latency compared with the existing conventional MQTT model. The simulation results expressed that proposed VEnvMQTT model achieves the ~20% reduced latency compared with the conventional MQTT model.

References

[1] Mehmood, M. Y., Oad, A., Abrar, M., Munir, H. M., Hasan, S. F., Muqeet, H., &Golilarz, N. A. (2021). Edge computing for IoT-enabled smart grid. *Security and Communication Networks*, *2021*.

[2] Yazid, Y., Ez-Zazi, I., Guerrero-González, A., El Oualkadi, A., &Arioua, M. (2021). UAV-enabled mobile edge-computing for IoT based on AI: A comprehensive review. *Drones*, *5*(4), 148.

[3] Abrar, M., Ajmal, U., Almohaimeed, Z. M., Gui, X., Akram, R., &Masroor, R. (2021). Energy efficient UAV-enabled mobile edge computing for IoT devices: a review. *IEEE Access*.

[4] Zhou, X., Yang, X., Ma, J., Kevin, I., & Wang, K. (2021). Energy efficient smart routing based on link correlation mining for wireless edge computing in IoT. *IEEE Internet of Things Journal*.

[5] Raj, J. S., & Jennifer, S. (2021). Optimized mobile edge computing framework for IoT based medical sensor network nodes. *Journal of Ubiquitous Computing and Communication Technologies (UCCT)*, *3*(01), 33-42.

[6] Savaglio, C., &Fortino, G. (2021). A simulation-driven methodology for IoT data mining based on edge computing. *ACM Transactions on Internet Technology (TOIT)*, *21*(2), 1-22.

[7] Yar, H., Imran, A. S., Khan, Z. A., Sajjad, M., &Kastrati, Z. (2021). Towards smart home automation using IoT-enabled edge-computing paradigm. *Sensors*, *21*(14), 4932.

[8] Sheng, S., Chen, P., Chen, Z., Wu, L., & Yao, Y. (2021). Deep reinforcement learning-based task scheduling in iot edge computing. *Sensors*, *21*(5), 1666.

[9] Chen, X., Li, M., Zhong, H., Ma, Y., & Hsu, C. H. (2021). DNNOff: offloading DNN-based intelligent IoT applications in mobile edge computing. *IEEE transactions on industrial informatics*, *18*(4), 2820-2829.

[10] Li, X., Zhao, L., Yu, K., Aloqaily, M., &Jararweh, Y. (2021). A cooperative resource allocation model for IoT applications in mobile edge computing. *Computer Communications*, *173*, 183-191.

[11] Munir, M. S., Bajwa, I. S., Ashraf, A., Anwar, W., & Rashid, R. (2021). Intelligent and smart irrigation system using edge computing and IoT. *Complexity*, *2021*.

[12] Vaiyapuri, T., Parvathy, V. S., Manikandan, V., Krishnaraj, N., Gupta, D., & Shankar, K. (2021). A novel hybrid optimization for cluster-based routing protocol in information-centric wireless sensor networks for IoT based mobile edge computing. *Wireless Personal Communications*, 1-24.

[13] Zhan, C., Hu, H., Liu, Z., Wang, Z., & Mao, S. (2021). Multi-UAV-enabled mobile-edge computing for time-constrained IoT applications. *IEEE Internet of Things Journal*, *8*(20), 15553-15567.

[14] Islam, J., Kumar, T., Kovacevic, I., &Harjula, E. (2021). Resource-aware dynamic service deployment for local iot edge computing: Healthcare use case. *IEEE Access*, *9*, 115868-115884.

[15] Nguyen, P. X., Tran, D. H., Onireti, O., Tin, P. T., Nguyen, S. Q., Chatzinotas, S., & Poor, H. V. (2021). Backscatter-assisted data offloading in OFDMA-based wireless-powered mobile edge computing for IoT networks. *IEEE Internet of Things Journal*, *8*(11), 9233-9243.

[16] Huong, T. T., Bac, T. P., Long, D. M., Thang, B. D., Binh, N. T., Luong, T. D., &Phuc, T. K. (2021). Lockedge: Low-complexity cyberattack detection in iot edge computing. *IEEE Access*, *9*, 29696-29710.

[17] Borsatti, D., Davoli, G., Cerroni, W., &Raffaelli, C. (2021). Enabling industrial IoT as a service with multi-access edge computing. *IEEE Communications Magazine*, *59*(8), 21-27.

[18] Singh, A., Satapathy, S. C., Roy, A., &Gutub, A. (2022). Ai-based mobile edge computing for iot: Applications, challenges, and future scope. *Arabian Journal for Science and Engineering*, 1-31.

[19] Simpson, S. V., & Nagarajan, G. (2021). A fuzzy based co-operative blackmailing attack detection scheme for edge computing nodes in MANET-IOT environment. *Future Generation Computer Systems*, *125*, 544-563.