Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 3, July 2026
DOI : https://doi.org/10.33395/sinkron.v10i3.16373
e-ISSN : 2541-2019
p-ISSN : 2541-044X

A Regional Edge Cluster Approach for Reliable Connectivity in Mobile Healthcare IoT Systems

Ali AL-ALLAWEE[1)*], Senan A. M. Alhasan[2)]
[1)2)]Department of Computer Science, College of Education for Pure Science, University of Mosul, Mosul, Iraq.
[1)]aliabd@uomosul.edu.iq [2)]senan.alhasan@uomosul.edu.iq

Submitted :June 12, 2026 | **Accepted** : July 11, 2026 | **Published** : July 13, 2026

Abstract: Healthcare applications are known to be critical and time-sensitives nowadays. Recently, several healthcare applications have been deployed utilizing Internet of Things (IoT) technologies due to its efficiency, management, and mobility which yield accordingly to enhance patients' health. Several researches have been achieved into IoT-healthcare and Edge computing to enhance better mobility and near processing. However, patient mobility, especially with high speed transportation and intersection, leads IoT-healthcare devices to loss connections and process with latency. This research proposes a Patient Mobility Control under Edge Computing (PMC-EC) mechanism that aims to guaranty to support connectivity between patients (IoT Health sensor) and clouds by using edge computing paradigm in middle. We collected the results by utilizing cluster mechanisms which arranged between clouds/ edge computing servers. The mechanism is implemented in the platform of OpenStack cloud server and been evaluated by calculating the latency and throughput parameters in two different scenarios. The results, in sum, are collected with promised indication of latency and throughput values to provide acceptable performance.

Keywords: Edge Computing, Cloud, IoT Healthcare, Cluster.

INTRODUCTION

Cloud computing is an evolution of information-technology in providing IT resources remotely. In recent, Cloud computing had gained much consideration in either research and industry sectors (Donnoet et al., 2019) (Cao et al., 2020). Furthermore, cloud computing enables end-users to access computing resources on demand such as storage, memory, processing power, applications, and even complete infrastructure, through network connectivity. That makes our daily live depends on cloud computing widely by storing backups, writing and/or editing online documents, collaborating business information, and playing games online(Wang et al., 2020) (Kolhar et al., 2017). However, in certain applications, cloud still far from the desired requirements in term of communication, latency, and response time especially with a real time IoT applications such as healthcare and vehicle (Ahvar et al., 2022). That is due to the nature structure of cloud been implemented in far network. Edge computing is another paradigm found to sit between clouds and IoT sensors (Ren et al., 2019).

Edge computing is an extension thought of cloud computing, taht share almost same characteristics with cloud computing (Nayyer et al., 2022) (Du et al., 2024). In cloud computing, the IoT devices always try to migrate their high processes, depth analysis, and power-energy to the nearest edge node. Some important data processing in real-time, including decision making are handled in edge server rather than far cloud. In the instance, edge servers operate to share its current information with the data center with its corresponding cloud periodically (Zhao et al., 2024). Unlike cloud computing, which primarily provides centralized processing, in fact, edge computing emphasizes localized operations, small data, and real-time intelligent analysis. As a result, data is operated and to be stored later closer to its source. That will minimize unnecessary data transmission to the cloud. In general, this will improves most network parameters such as bandwidth utilization, response time, and low latency. (Alanhdi and Toka, 2024).

Generally, the place of the edge computing is located closer to the data source (IoT sensors). Such case stores most frequent retrieved data at the edge and some of the operations will carried out in the edge computing node as well. This will decrease the in-between data transmission operations and communication. As a results, IoT users will gain high reliable intelligent services, efficient data-transmission, and low overall latency. One of the

*name of corresponding author

important sector that get advantage of the discussed features in edge computing is healthcare services. Healthcare, requires high accurate, speed, and scalable systems in today's technologies.

E-health applications have been deployed in different aspects in health sector including emergency alerts, early issue detection, and computer-assisted rehabilitation (Amutha, 2023). The IoT healthcare system provides an efficient monitoring and tracking for end-device/patient. That can be achieved by utilizing the cloud/edge computing services with resource sharing, flexibility, data integration, parallel processing and tracking (Jiang et al., 2023) (Mpembele et al., 2023). However, delay and lost connection from the IoT device (patients) to the far-end cloud servers considered to be serious issue in healthcare application especially during patient mobility.

This paper proposes a mechanism to improve and maintain reliable connectivity between patients equipped with IoT healthcare sensors and cloud services by incorporating an edge computing layer. The proposed approach make use of edge servers to provide uninterrupted communication for patients while they are traveling across different geographical regions (such as high-speed transportation). To achieve this, the mechanism continuously monitors patient mobility and dynamically associates each patient with the most appropriate nearby edge server by keep synchronizing the patient's vital information among collaborating edge servers.

The proposed system is implemented using an OpenStack-based cloud platform connected to three edge servers. Its performance is evaluated under two experimental scenarios and evaluated through throughput and latency as a main performance metrics. The objective is to ensure seamless connectivity while maintaining acceptable communication performance during patient mobility.

The main contributions of this work are summarized as follows:

1. A collaborative edge server architecture based on regional cluster management to support continuous patient connectivity.
2. The implementation and evaluation of a regional edge computing infrastructure integrated with an OpenStack cloud environment.

The remainder of this paper is organized as follows. Section 2 reviews the background and related work on edge computing. Section 3 describes the proposed mechanism. Section 4 presents the system implementation. Section 5 outlines the experimental methodology, while Section 6 discusses the performance evaluation and results. Finally, Section 7 concludes the paper and suggests directions for future work.

.

EDGE COMPUTING

Unlike traditional cloud computing, edge computing is a distributed computing paradigm that performs data processing and computation closer to the network edge. The general aim is to provide all kind of computing closer to the source of the data (Cao et al., 2020) (Satyanarayanan, 2017). Edge computing is defined by several scientific and researchers thoughts in variant domains in the field. For instance, a definition by Shi et al. ''Edge computing is a new computing mode of network edge execution. The downlink data of edge computing represents cloud service, the uplink data represents the Internet of Everything, and the edge of edge computing refers to the arbitrary computing and network resources between the data source and the path of cloud computing center.'' (Shi et al., 2019) (Shi et al., 2017). Also, professor M. Satyanarayanan defines edge computing as: ''Edge computing is a new computing model that deploys computing and storage resources (such as cloudlets, micro data centers, or fog nodes, etc.) at the edge of the network closer to mobile devices or sensors'' (Satyanarayanan, 2017). China edge computing industry defines edge computing as: ''near the edge of the network or the source of the data, an open platform that integrates core capabilities such as networking, computing, storage, applications, and provides edge intelligent services nearby to meet the industry agility key requirements in connection, real-time business, data optimization, application intelligence, security and privacy'' (Hong and Wang, 2018). In other words, edge computing is operating to migrate partial of the task assigned to cloud's network toward the edge of the network which is very near to patients (IoT source). Whereas the second leg, IoT sensors' devices direct the heavy operations, power, and fast decision right to the same related edge computing. See the Figure (1) which compares patient's connections with cloud vs. edge computing. In this paper the edge computing is utilized to ensure the scalability of patient mobility by make use of the cluster of edge server. In such case, patients reduce connections with the cloud.

*name of corresponding author

e-ISSN : 2541-2019
p-ISSN : 2541-044X

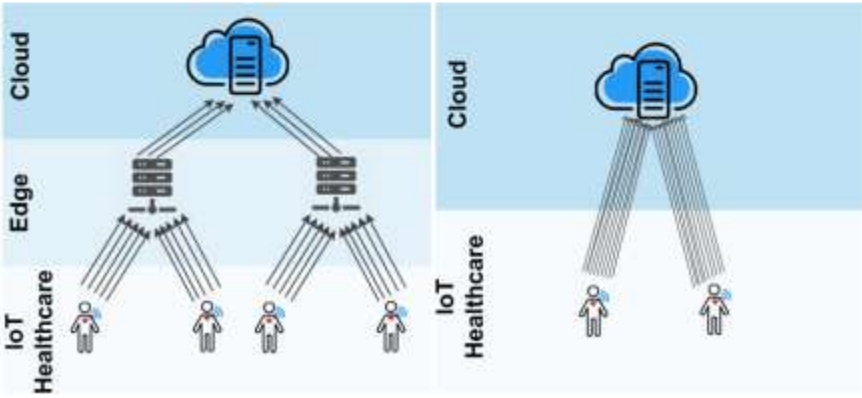

Fig. 1 Edge vs. cloud server

IOT HEALTHCARE

This section reviews the most recent research contributions that address the identified challenges in the field of IoT healthcare. In general, it summarizes the existing solutions which all proposed by previous studies and highlights the strengths and limitations. Where appropriate, figures, diagrams, and flowcharts are included to illustrate the proposed approaches and facilitate a clearer understanding of the underlying concepts and methodologies. In healthcare, smart IoT medical devices connect patients and smart objects to provide simple and healthy life (Mahmud et al., 2018). Specifically, Internet of Medical Things (IoMT) or Internet of Things Healthcare (IoTH) considered to be a vital part in healthcare domain. They provide smart services for healthcare by generating various data types and transmitting upstream to cloud/edge repositories. Medical devices can be fixed in variant location such as hospital, clinic, or installed on patient's body. Devices installed on patient's body are designed to provide services such as remote monitoring, heart rate, blood pressure, etc. These techniques help the strategies of patient medical treatment with an efficient way (Besher et al., 2022).

Other interesting part of smart healthcare architecture is the remote operation which also considered to be a revolution transmission in health domain. This technique required very high performance, accurate data delivery, and stable network to be achieved successfully. However, this kind of healthcare techniques do not covered in this research as we only consider IoT device mobility.

Generally speaking, most of IoT healthcare device are generating a very large amount of data streaming, it can be 10 GB per day per device. Such huge amount of data needs to be handled with high efficient way and performance.

RELATED WORK

This section presents the state of the art of how IoT healthcare mobiles within edge computing realm. The authors in (Abdelmoneem et al., 2020) contribute to the domain by presenting most related IoT architecture, allocation protocols, and mobility for connected devices of healthcare. They also proposed mechanism to enhance the mobility of the device of the patients via an adaptive Received Signal Strength (RSS) by utilizing the handoff technique. The proposed architecture enables dynamic distribution of healthcare workloads across cloud and fog computing servers. To realize this objective, the authors developed a mobility aware heuristic scheduling and resource allocation approach, referred to as MobMBAR. The proposed method optimizes task placement by considering patient mobility patterns together with the spatial characteristics of the collected sensor data, thereby improving load balancing and resource utilization across the computing infrastructure. The objective of the proposed approach is to decrease total schedule time using the most operation features such as; higher response time and critical level of individual task during reallocation phase. The authors work have been validated the results by compared with existing solution using simulation.

(Anwesha et al., 2021) proposed a mobile healthcare framework based on a collaborative edge–fog–cloud architecture. In this framework, edge and fog nodes are responsible for real-time health monitoring and preliminary data processing, while cloud servers perform advanced analytics, including the detection of abnormal health conditions. However, maintaining service continuity during patient transportation remains a significant challenge. Frequent changes in a patient's location can disrupt communication and delay the delivery of critical healthcare services, which may have serious consequences in emergency situations. Because the delay in delivering health data and the ongoing connection. The authors have claimed the consideration of mobility setups of the users with detection of the cloud. This helps to promote users with closer health center. Results found by this research and after theoretical analysis, they observed that the proposed framework lower the energy and delay of IoT device. In

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 3, July 2026
DOI : https://doi.org/10.33395/sinkron.v10i3.16373
e-ISSN : 2541-2019
p-ISSN : 2541-044X

addition, experimental analysis had showed that the proposed framework operated with recall value, time-efficiency, and high-precision compared to existing models.

The work in (Hosono et al., 2022) has the same idea of smart device mobility but with vehicle environment. In this study, the authors partitioned the roadway into multiple lane section IDs, with each section being managed by a designated edge server. This approach was implemented using a dynamic mapping system to enable vehicles to maintain continuous connectivity with the moving across different edge server regions. By coordinating communication among multiple edge servers the proposed system minimizes the impact of unstable wireless signals and facilitates reliable vehicle data aggregation. The performance evaluation primarily examined the effectiveness of the proposed approach in terms of load balancing and scalability of dynamic mapping system across multiple edge servers.

METHOD

We illustrate the design of the proposed mechanism by describing the premium actions that involved in the methodology steps to obtain the fourth mentioned objectives of this research. The general feature of PMC-EC mechanism is to deliver stable and smooth mobility for patient who moves (in high speed) from domain to the next one within the range area of edge computing servers. In addition, edge server cooperates to smoothen patient's mobility by sharing the most mobility related information priorly. The PMC-EC mechanism develops the cluster management to facilitate the process of sharing cluster-information between edge computing servers (Al-Allawee et al., 2023) (Lorenz et al., 2023). This mechanism ensures smooth connectivity and latency-low for patient's IoT-device traffic streaming, more specifically in high-speed mobility (Abualhaj et al., 2024) (Praptodiyono et al., 2023). The main steps are explained in details in the following:

1. At the beginning of the process, every edge server (cluster node) registers with the cloud coordinator (cluster head). In this registration, the cluster head builds and maintains a topology map containing the geographical locations of all edge nodes. Also, it continuously updates this topology whenever an edge server joins or leaves the cluster.
2. After a patient establishes a connection with an edge server of the currently serving node transfers the patient's relevant information to the most likely next edge server along the patient's movement path. In this step, the previous edge server temporarily retains the patient's context to ensure continuous tracking of the handover process.
3. The current edge server then reports the patient's updated location to the cloud to become able to estimate the patient's subsequent destination and support proactive resource preparation.
4. As the patient enters the coverage area of the next edge server, the same sequence of operations is repeated starting from Step 2.

This workflow enables patient information to be transferred, prepared, and processed at the anticipated destination edge server before the patient arrives. As a result from these steps, latency is minimized beside the reduce the additional connection establishment. Figure 2 presents the main components of the proposed mechanism.

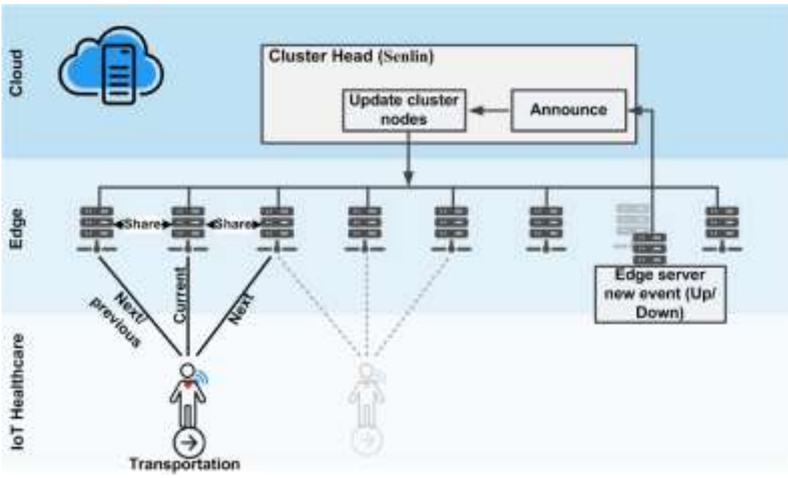

Fig. 2 PMC-EC mechanism components.

*name of corresponding author

IMPLEMENTATION

This section shows the implementation part of this work by describing the lab setup generally. In this research we utilized OpenStack (Bumgardner et al., 2016) as cloud and edge computing concept. OpenStack is an open-source platform to develop and configure cloud components and entities. The general function of OpenStack is to create and manage the computer system infrastructure such as network, storage, and other resources. OpenStack also found to support other services and stable platforms for end-user or company. In this research, to make the cluster operates, we setup the Senlin clustering service (Bumgardner et al., 2016), The cluster infrastructure was deployed and orchestrated using OpenStack Senlin. First, a cluster profile defining the characteristics of the compute nodes and it is created with the command openstack cluster profile create myserverProf. This profile is to be used to instantiate a new cluster through the command senlin cluster-create -p myserverProf clusterName. After the cluster was created, the main capacity was adjusted to three edge nodes using senlin cluster-resize --capacity 3 clusterName.

To keep the experimental environment simple and focus on validating the proposed mechanism which to focus on both the cloud server and all edge nodes. This configuration is all defined and setup in default OpenStack profile without additional customization. This minimized configuration overhead and eliminated the influence of advanced optimization settings to allow the evaluation to concentrate solely on demonstrating the feasibility of the proposed approach.

The proposed implementation also employs the OpenStack Receiver service to detect events generated when a patient connects to an edge node. Upon receiving such an event, application programming interfaces (APIs) are invoked to transfer the patient's context such as patient_ID, from the currently serving edge node to the predicted next node. Figure (3) illustrates the sequence of message exchanges among the cloud and edge network components during this process.

TESTING ENVIRONMENT

This section shows the implementation part of this work by describing the lab setup generally. In this research we utilized OpenStack (Bumgardner et al., 2016) as cloud and edge computing logic. The testing environment of this mechanism is accomplished by illustrating the methodology of the test, the source of the traffic load with evaluation parameters. Devices (such as cloud, edges, IoT Healthcare) included in this testing environment are all connected with local area (wired). The evaluation omit some clouds' functionalities for instances such as create or delete VMs, write to the desk, and manage the VMs. Which is in fact only sends normal string and the to the sink. However, the evaluation conducted in this study focuses on examining the connection workflow among cluster nodes to demonstrate the feasibility of the proposed mechanism and validate the underlying concept.

The platform used in this research for testing uses VMTP (VMTP et al., 2024) as a tool measurement as data path performance evaluator is built specifically for OpenStack clouds. The VMTP is designed to test the connection path between VMs in different scenarios. Here we have chosen two different path scenarios tailurated to research concern which are:

- **Flow 1 (East-West Communication):** This communication path occurs between virtual machines (VMs) residing on the same private network and using fixed private IP addresses. It represents the communication channel between the edge node and the cloud node, where both endpoints maintain static IP assignments.
- **Flow 2 (East-West Communication):** This communication path takes place between VMs located on different private networks while still using fixed IP addresses. It models the interaction between a mobile patient and the serving edge node as the patient moves across different network domains.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 3, July 2026
DOI : https://doi.org/10.33395/sinkron.v10i3.16373
e-ISSN : 2541-2019
p-ISSN : 2541-044X

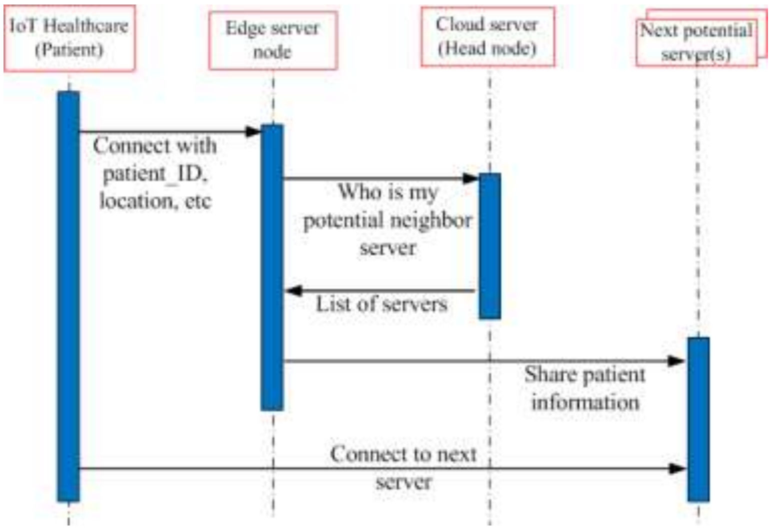

Fig. 3 Passing messages

We can notice the patient IoT healthcare device transmitting any kind of data toward the edge computing node, despite which application does he/she utilize.

Scenario 2 starts to be invoked the time that the patient transmit from current edge server to the upcoming next server, which operates manually, as the network mobility and handover is not the main scope of this research.

The proposed mechanism was assessed according to the most critical parameters is round-trip-time (*rtt*) in ms for latency measurement, calculated in (1). Throughput (tp) measured in kB/s which calculated in (2). For the entire test, the assessment is applied with 500 concurrent load connections to meet the overall test goal.

$$AvgLatency = \frac{RTT_{\{s1\}}+RTT_{\{s2\}}}{2} + \frac{RTT_{\{c1\}}+RTT_{\{c2\}}}{2} \quad (1)$$

$$TH = \frac{I}{T} + \frac{RTT_{\{c1\}}+RTT_{\{c2\}}}{2} \quad (2)$$

where (RTT) denotes the round-trip time, (s) represents the server, and (c) represents the client. For the throughput calculation, (I) denotes the amount of transmitted data (inventory), and (T) represents the elapsed transmission time.

RESULTS

This section present the most noticeable and interesting results from the evaluation phase. This section presents the key findings and significant outcomes of the proposed research. The performance evaluation was conducted under two experimental scenarios, corresponding to the two communication stages of the proposed mechanism. Scenario 1 evaluates the communication performance between the patient's IoT devices and edges computing server. Scenario 2 examines the communication between edge servers and the cloud platform.

Scenario 1: Figure (4) clearly describes the latency in high performance regarding patient IoT device stable to transfer data toward edge server, the data transmitted is spends approximately 122 ms in terms of latency. The test going to increase with 16% as the load increases. The stability is still in good condition due to no operations are processed with a stable procedures. From that, we can clearly notice the proposed PMC-EC mechanism spends 160 ms when 100 load. Also, we can notice that Latency exhibited an increasing trend with higher workloads, rising to 280 ms at a load of 200 and 370 ms at a load of 300. This latency increase is primarily caused by the edge server processing a larger number of simultaneous connections, which increases its workload of the cloud and other neighbor edge server. In general, the patient device expected to encounter more latency in the case that it mobile to next domain. This improvement is achieved by utilizing the preprocessed information that has already been shared by the previous edge server.

Figure (5) illustrates the throughput performance of the proposed mechanism done with different operating conditions. Obviously, from the normal communication phase, the edge server receives data from the IoT device at an average throughput of 56.2 kB/s with the workload of 100, this can show efficient communication performance. We also notice that this relatively high throughput is achieved because the edge server primarily communicates with the IoT device and not heavily engaged in coordination with other cluster nodes.

When the patient moves to the coverage area of another edge server the PMC-EC mechanism is activated to support service continuity. If we look to handover process, the measured throughput decreases to approximately

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 3, July 2026
DOI : https://doi.org/10.33395/sinkron.v10i3.16373
e-ISSN : 2541-2019
p-ISSN : 2541-044X

52% of the ideal value. Thus, this reduction is mainly attributed to the additional communication and synchronization overhead incurred whereas the current edge server exchanges information with both the cloud platform and the destination edge server to maintain uninterrupted patient connectivity.

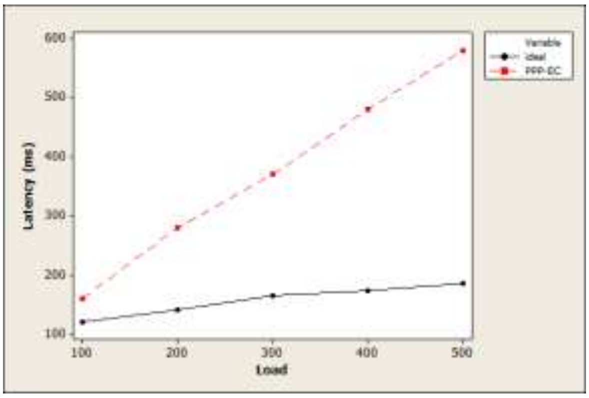

Fig. 4 Round-trip latency between the IoT device and the edge server under Scenario 1.

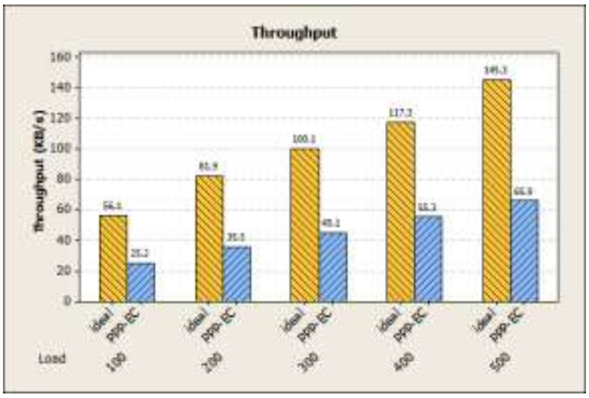

Fig. 5 Throughput between the IoT device and the edge server under Scenario 1.

Scenario 2: The results collected from this scenario is the interesting part because the calculation is from the the edge to cloud server. In Figure (6), present the total time collapse to manage cluster nodes as the patient transforms to upcoming server. Under a workload of 100, the proposed mechanism records a latency of 144 ms, which is approximately 90 ms higher than that observed under the ideal operating condition. Even though this increase, the latency remains within an acceptable range, as it is well below the one-second threshold required for maintaining responsive communication. This increase can be attributed to the additional communication overhead associated with synchronizing data with the cloud, exchanging information with the destination edge server, and retrieving the required records from the database. Also, we noticed that in the 500 load test, the PMC-EC tooks 498 ms which shows more variant from ideal scenario. This is true since there is still extra intersection between neighbor of region servers. The intersection might occurred with the same server before.

Figure (7) presents the throughput performance of the proposed PMC-EC mechanism during communication between the edge server, the cloud platform, and neighboring edge servers. Under normal operating conditions we can notice the edge server transmits a relatively small amount of data to the cloud. This can achieve a throughput of 25.2 kB/s under a workload of 100. This low throughput is expected because no major synchronization or cluster management operations are performed during this stage. The observed traffic primarily consists of routine background activities like cluster status monitoring, node join or leave events, and policy synchronization.

As the workload increases, additional communication is required to support cluster management operations. At a workload of 500 the throughput increases by approximately 57 kB/s compared with the ideal operating condition. This increase represents the communication overhead associated with cluster activities. Furthermore, the proposed system is deployed within a single local network using fixed IP addresses, then minimizing the impact of external network conditions and ensuring that the measured performance primarily reflects the behavior of the proposed mechanism.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 3, July 2026
DOI : https://doi.org/10.33395/sinkron.v10i3.16373
e-ISSN : 2541-2019
p-ISSN : 2541-044X

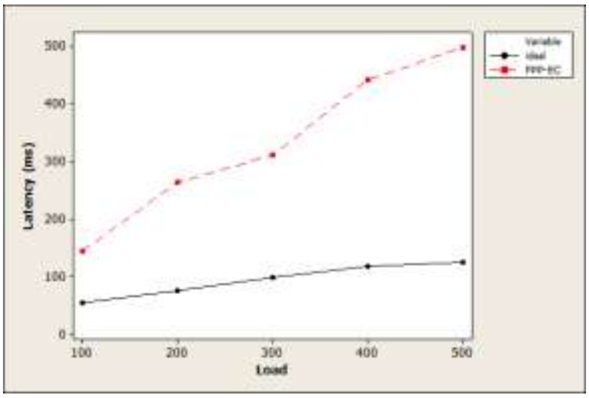

Fig. 6 Round-trip latency between the edge server and the cloud platform under Scenario 2.

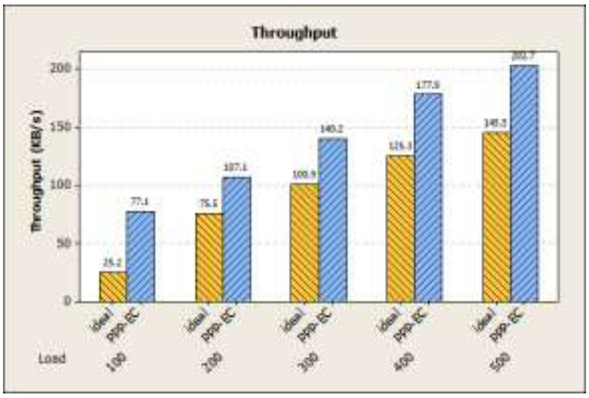

Fig. 7 Throughput between the edge server and the cloud platform under Scenario 2.

CONCLUSION

In recent, approximately 10 billion IoT healthcare devices operated today, such as heart rate monitors, insulin pumps, and pacemakers. These devices generate a plenty of data continuously to be processed and analyzed for clinicians provide better care. In healthcare IoT systems, the data collected by medical sensors must be transmitted to a computing platform for processing and analysis, traditionally the cloud. However, the centralized nature of cloud computing places processing resources far from patients, which can introduce communication delays. Since healthcare applications often require rapid and accurate real-time analysis, relying solely on distant cloud servers may not satisfy stringent latency requirements. Consequently, processing data closer to patients through edge computing provides faster response times and improves the delivery of time-critical healthcare services. Thus, edge computing is wise choose in healthcare which provide computing power and near data process to IoT healthcare sensors. However, while patient transformation, the patient serves with slow connectivity especially when transform from domain to next domain. This work proposed a mechanism that guaranty and enhance the connectivity between patients (IoT Healthcare sensors) and edge computing servers. Edge point operates keeps deliver better connection for patient who is moving among regions with high speed transportation. The proposed mechanism is implemented and assessed with OpenStack platform as cloud server along with three edge server clusters. Results have been collected by calculating latency and throughput for all mechanisms in scenario 1 and 2. N sum, the proposed mechanism is achieved with stable performance with 160 ms when 100 load, the latency increase as the load increase to spend 280 and 370 ms in (200) and (300) load respectively. The valued of results are acceptable and valid in such latency percentage. In addition, scenario 2 have presented the same perspective when the 500 load is applied the throughput with 57 kB/s difference from the ideal status. Finally, future work is plan to deploy some security perspectives while sharing sensitive patient's data and concern topology hiding paradigm.

REFERENCES

Abdelmoneem, R. M., Abderrahim, B., & Eman, S. (2020). Mobility-aware task scheduling in cloud-fog IoT-based healthcare architectures. *Computer Networks, 179*, Article 107348.

Abualhaj, M., Al-Zyoud, M., Hiari, M., Alrabanah, Y., Anbar, M., Amer, A., & Al-Allawee, A. (2024). A fine-tuning of decision tree classifier for ransomware detection based on memory data. *International Journal of Data and Network Science, 8*(2), 733–742.

*name of corresponding author

Ahvar, E., Orgerie, A.-C., & Lebre, A. (2022). Estimating energy consumption of cloud, fog, and edge computing infrastructures. *IEEE Transactions on Sustainable Computing, 7*(2), 277–288. https://doi.org/10.1109/TSUSC.2019.2905900

Al-Allawee, A., Lorenz, P., Abouaissa, A., & Abualhaj, M. (2023). A performance evaluation of in-memory databases operations in session initiation protocol. *Network, 3*(1), 1–14.

Al-Allawee, A., Mihoubi, M., Lorenz, P., & Abakar, K. S. (2023). Efficient dispatcher mechanism for SIP cluster based on memory utilization. In *Proceedings of the IEEE International Conference on Communications (ICC 2023)* (pp. 3370–3375). IEEE. https://doi.org/10.1109/ICC45041.2023.10278652

Alanhdi, A., & Toka, L. (2024). A survey on integrating edge computing with AI and blockchain in maritime domain, aerial systems, IoT, and Industry 4.0. *IEEE Access, 12*, 28684–28709. https://doi.org/10.1109/ACCESS.2024.3367118

Amutha, B. (2023). IoT revolutionizing healthcare: A survey of smart healthcare system architectures. In *2023 International Conference on Research Methodologies in Knowledge Management, Artificial Intelligence and Telecommunication Engineering (RMKMATE)* (pp. 1–5). IEEE. https://doi.org/10.1109/RMKMATE59243.2023.10369980

Anwesha, M., Shreya, G., Aabhas, B., Soumya, K., & Buyya, R. (2021). Internet of Health Things (IoHT) for personalized health care using integrated edge-fog-cloud network. *Journal of Ambient Intelligence and Humanized Computing, 12*, 943–959.

Besher, K. M., OKidhain, I., Wick, L., & Ali, M. Z. (2022). Congestion control of healthcare packet routing in 5G edge computing networks. In *2022 International Conference on Engineering and Emerging Technologies (ICEET)* (pp. 1–6). IEEE. https://doi.org/10.1109/ICEET56468.2022.10007159

Bumgardner, C. (2016). *OpenStack in action*. Manning Publications.

Cao, K., Liu, Y., Meng, G., & Sun, Q. (2020). An overview on edge computing research. *IEEE Access, 8*, 85714–85728. https://doi.org/10.1109/ACCESS.2020.2991734

De Donno, M., Tange, K., & Dragoni, N. (2019). Foundations and evolution of modern computing paradigms: Cloud, IoT, edge, and fog. *IEEE Access, 7*, 150936–150948. https://doi.org/10.1109/ACCESS.2019.2947652

Du, J., Zhang, G., Yuan, X., & Zang, X. (2024). P²SPA: Privacy preservation strategy with pseudo-addresses for edge computing networks. *IEEE Access, 12*, 40962–40972. https://doi.org/10.1109/ACCESS.2024.3377102

Hong, X., & Wang, Y. (2018). Edge computing technology: Development and countermeasures. *Chinese Journal of Engineering Science, 20*(2), 20.

Hosono, K., Maki, A., Watanabe, Y., Takada, H., & Sato, K. (2022). Implementation and evaluation of load balancing mechanism with multiple edge server cooperation for dynamic map system. *IEEE Transactions on Intelligent Transportation Systems, 23*(7), 7270–7280. https://doi.org/10.1109/TITS.2021.3067909

Jiang, L.-W., Li, Y.-J., Lin, L.-S., & Kao, C.-H. (2023). Sleep quality monitoring IoT-based healthcare system. In *2023 IEEE 5th Eurasia Conference on Biomedical Engineering, Healthcare and Sustainability (ECBIOS)* (pp. 92–95). IEEE. https://doi.org/10.1109/ECBIOS57802.2023.10218456

Kolhar, M., Abu-Alhaj, M. M., & Abd El-atty, S. M. (2017). Cloud data auditing techniques with a focus on privacy and security. *IEEE Security & Privacy, 15*(1), 42–51. https://doi.org/10.1109/MSP.2017.16

Mahmud, R., Koch, F. L., & Buyya, R. (2018). Cloud-fog interoperability in IoT-enabled healthcare solutions. In *Proceedings of the 19th International Conference on Distributed Computing and Networking* (pp. 1–10).

Mpembele, A. B., Rogers, T., Ghosh, U., & Shetty, S. (2023). Communication-efficient and privacy-preserving edge-cloud framework for smart healthcare. In *2023 IEEE Globecom Workshops (GC Wkshps)* (pp. 377–382). IEEE. https://doi.org/10.1109/GCWkshps58843.2023.10464501

Nayyer, M. Z., et al. (2022). LBRO: Load balancing for resource optimization in edge computing. *IEEE Access, 10*, 97439–97449. https://doi.org/10.1109/ACCESS.2022.3205741

Praptodiyono, S., Firmansyah, T., Anwar, M. H., Wicaksana, C. A., Pramudyo, A. S., & Al-Allawee, A. (2023). Development of hybrid intrusion detection system based on Suricata with pfSense method for high reduction of DDoS attacks on IPv6 networks. *Eastern-European Journal of Enterprise Technologies, 125*(9), 75–84.

*name of corresponding author

DOI : https://doi.org/10.33395/sinkron.v10i3.16373
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Ren, J., Yu, G., He, Y., & Li, G. Y. (2019). Collaborative cloud and edge computing for latency minimization. *IEEE Transactions on Vehicular Technology, 68*(5), 5031–5044. https://doi.org/10.1109/TVT.2019.2904244

Satyanarayanan, M. (2017). The emergence of edge computing. *Computer, 50*(1), 30–39. https://doi.org/10.1109/MC.2017.9

Shi, W. S., Zhang, X. Z., & Wang, Y. F. (2019). Edge computing: State-of-the-art and future directions. *Journal of Computer Research and Development, 56*(1), 1–21.

Shi, W., Sun, H., Cao, J., Zhang, Q., & Liu, W. (2017). Edge computing—An emerging computing model for the Internet of Everything era. *Journal of Computer Research and Development, 54*(5), 907–924.

VMTP. (2024, January 19). *VMTP is a data path performance measurement tool for OpenStack clouds*. https://vmtp.readthedocs.io/en/latest/readme.html

Wang, B., Wang, C., Huang, W., Song, Y., & Qin, X. (2020). A survey and taxonomy on task offloading for edge-cloud computing. *IEEE Access, 8*, 186080–186101. https://doi.org/10.1109/ACCESS.2020.3029649

Zhao, P., Yang, Z., & Zhang, G. (2024). Personalized and differential privacy-aware video stream offloading in mobile edge computing. *IEEE Transactions on Cloud Computing, 12*(1), 347–358. https://doi.org/10.1109/TCC.2024.3362355

*name of corresponding author