Performance Evaluation of an Internet of Healthcare Things for Medical Monitoring Using M/M/c/K Queuing Models

FRANCISCO AIRTON SILVA, TUAN ANH NGUYEN, IURE FÉ, CARLOS BRITO, DUGKI MIN, AND JAE-WOO LEE

1Federal University of Piauí (UFPI), Teresina 560 054, Brazil
2Konkuk Aerospace Design-Airworthiness Research Institute (KADA), Konkuk University, Seoul 05029, South Korea
3Brazilian Army, Picos 64606-000, Brazil
4Department of Computer Science and Engineering, College of Engineering, Konkuk University, Seoul 05029, South Korea
5Department of Aerospace Engineering, Konkuk University, Seoul 05029, South Korea

Corresponding author: Dugki Min (dkmin@konkuk.ac.kr)

This work was supported in part by the Brazilian National Council for Scientific and Technological Development—CNPq under Grant 309335/2017-5, in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant 2020R1A6A1A03046811, in part by the Ministry of Science, ICT (MSIT), South Korea, through the Information Technology Research Center (ITRC) Support Program supervised by the Institute for Information & Communications Technology Planning & Evaluation (IITP) under Grant IITP-2020-2016-0-00465, and in part by “The Competency Development Program for Industry Specialist” of the Korean Ministry of Trade, Industry, and Energy (MOTIE), operated by the Korea Institute for Advancement of Technology (KIAT) under Grant N0002428.

ABSTRACT Due to the non-stop and rapid spreading of virus pandemics all over the world, traditional healthcare monitoring capabilities of hospitals and/or medical centers are under a severe over-load. Modern computing infrastructures with the harmony of various layers of computing paradigms (e.g., cloud/fog/edge computing) for healthcare monitoring are apparently the essential computing backbone that help access and process instantly the medical data of every single patient at the very edge of the healthcare system to combat with global or regional virus contagion. Previous studies proposed different computing system architectures for healthcare monitoring but few works considered the evaluation of pure performance of medical data transmission in a comprehensive manner. In this paper, we proposed an M/M/c/K queuing network model for the performance evaluation of an Internet of Healthcare Things (IoHT) infrastructure in association with a three layer cloud/fog/edge computing continuum. The model considers a life cycle of medical data from body-attached IoT sensors in edge layer all the way to local clients (e.g., local medical doctors, physicians) through fog layer and to remote clients (e.g., medical professionals, patient’s family members) through cloud layer. Furthermore, we also explore the impact of the alteration in system configuration and computing capability of computing layers in two scenarios on various performance metrics. Critical performance metrics related to quality of service are evaluated in a comprehensive manner, such as (i) mean response time of medical data transmission to fog (local) clients and to cloud (remote) clients, (ii) utilization of cloud/fog/edge computing layers, (iii) service throughput, (iv) number of medical messages in a period of time, and (v) drop rate. The simulation results pinpoint bottle-neck parameters and configurations of the IoHT infrastructure’s system architecture in relation to the frequency of medical data collection for health check of patients. Thus, the findings of this study can help improve medical administration in hospitals and healthcare centers and help design computing infrastructures in accordance for medical monitoring in the severe circumstances of virus pandemics.

INDEX TERMS Healthcare monitoring, internet of healthcare things, queuing model, performance evaluation, quality of service.

I. INTRODUCTION

The Internet of Things (IoT) is one of the great advances in modern technology. IoT devices are inter-communicable and can extract information about different environments. IoT devices are used, for example, to monitor patients in smart hospitals [1], to monitor smart homes [2], or to detect accidents in vehicular ad-hoc networks [3].

1Statista: https://tinyurl.com/y83kkser
There are many involved parameters and analytical models can be highly expensive and impractical with real patients. The incorporation of IoT in health care is broadly termed as IoHT (Internet of Healthcare Things) [7].

IoHT is an ecosystem of medical devices, such as smart healthcare and monitoring devices (e.g., smart pacemaker, smart blood glucose meter etc.) that have the ability to collect, analyze and transmit health data to healthcare systems through communication networks. A typical IoHT scenario involves smart healthcare devices that monitor the vital signs of focal patients, and transmit that data directly to remote servers differentiated basically by location and resources capacity. Cloud platforms are examples of common computing resources in this context [8], [9]. Cloud computing may be used to store and analyze health data, enabling automated decision making regarding interventions. However, closer computing resources are needed to provide even lower processing delays. Fog and edge computing are complementary layers that fills this gap of local processing. Fog [10] brings part of the cloud capacity closer to the IoT layer. Edge devices [11] connect IoT sensors to the fog and cloud layers. Such a Cloud-Fog-Edge architecture provides the processing support needed by the IoHT constrained devices. This way, IoHT technologies can contribute to improve performance, reduce costs and mitigate risk [12]–[17]. However, although the IoHT have been adopted to monitor people everywhere, there is a place where technology must be much more efficient due to its inherent criticality: the hospital.

A smart hospital is a concept that emerged as a result of rapid digitalization across the healthcare industries with the use of IoHT, data analytics, availability of personalized services, Artificial Intelligence (AI), and lastly the Cloud-Fog-Edge processing capacities [1], [18]. Smart hospitals offer new opportunities to provide fast and accurate responses by obtaining relevant information. This intelligent smart hospital’s network can receive data from several sources (hospital rooms), process data in a decentralized manner with higher digital computing resources to make smarter decisions. From this, intelligent recommendations, predictive analysis, or pattern detection can be made. However, the system requirements of smart hospitals are highly critical, including performance. The performance evaluation of smart hospital systems is essential to ensure their optimal operation. Evaluating the performance of IoHT supported by Cloud-Fog-Edge resources with real experiments can be highly expensive and impractical with real patients. There are many involved parameters and analytical models that can be useful in this context, making predictions based on probabilities.

Queuing networks [19] have become extremely popular in the applied stochastic modeling field for various applications, such as telecommunications, computers, manufacturing, and transportation. Queuing network models are more straightforward and more didactic, in addition to being well suited for evaluating performance. The modeling and evaluation of pure performance help focus on the probabilistic nature of user demands (workload) as well as internal state behaviors without considering failure and repair occurrences which are often very rare events in physical systems. Thus, a pure performance model help represent data transactions within a certain period of time to discover performance bottlenecks due to architecture design. Especially, in the case of IoHT for medical monitoring with the use of high-reliability equipment, a single variation of system architecture, operational configuration and/or parameters of devices/components apparently cause a sensitive impact on the performance of processing data and service delivery due to stream-like end-to-end data transactions from medical sensors surrounding patients at the edge of the network to display devices at the medical professionals’ working places. Pure performance analysis based on product form queuing network in these cases can consider different aspects in a quick manner such as resources capacity variation, sensor grouping by location, number of processing cores per machine. Also, various performance metrics such as throughput, drop rate, mean response time, response time distribution, and utilization can be analyzed in a comprehensive manner.

There are some related work that have evaluated healthcare systems using analytical models [9], [20]–[26]. Some of them have focused on security aspects [21], [22], [27], [28]. Other studies have focused on infrastructure availability [24], [25]. However, only one paper have observed all the Cloud-Fog-Edge layers [9], but such a work did not evaluate pure performance aspects. In this context, only two related works adopted queue models [21], [24]. Marcu et al. [21] failed to explore the potential of multiple processing layers and they have proposed a model too much simplified with only two queues. Elter et al. [24] used fog and cloud layer, however, with the limitation of measuring only the mean response time (MRT) of the system. Furthermore, all related work have failed in representing essential smart hospitals characteristics, such as: sensors grouped by location and mean response time calculated per layer.

In that sense, this paper proposes to evaluate IoHT systems supported by Cloud-Fog-Edge using a comprehensive queuing model of type M/M/c/K. Queuing networks are suitable for detecting performance bottlenecks due to shared resources in distributed systems [29]. An advantage of the queue models is the low complexity of the steady-state solutions (polynomial in the number of queues and customers), as they can be obtained as a product of the steady-state solution for each of the individual queues in the network [30]. Therefore, the contributions of this paper are the following:
A queuing model is presented for performance evaluation, as a useful tool for designers of IoHT scenarios based on Cloud-Fog-Edge resources to verify the performance of changes in the system even before they are implemented. The model permits to configure dozens of parameters depending on how many components the designer needs, including service time, transmission time, resource capacities, and queue sizes;

- Resource capacity analysis that serves as a practical guide for performance analysis in a IoHT monitoring scenario with the proposed model. The tested scenarios have included the variation of Cloud-Fog resources. A significant number of performance metrics were considered in this work, such as resource utilization, mean response time, drop rate, and throughput.

The remaining of this paper is organized as follows. Section II presents related work with an extensive table comparing the literature with our proposal. Section III describes the architecture considered for designing the model including all necessary assumptions. Section IV details the proposed analytical queuing model and its workflow explanation. Section V presents and analyzes the obtained simulation results. Finally, Section VI draws the main conclusions and suggestions for further works on the topic.

II. RELATED WORKS

This section presents works that have approaches more similar to the present work. Table 1 presents a comparison of the collected works, highlighting our proposal’s main differences. Eight studies were surveyed, ordered by year. The oldest one was from 2014. All the papers have in common the use of some analytical model (e.g., queuing, Markov chain, Petri net models) to evaluate an architecture for monitoring people with sensors. All works focus on health context, with the exception of the article [26], which does not specify such a context. Next, the works are discussed in a grouped manner according to each comparison criterion.

A. MAIN ARCHITECTURE COMPONENTS

Most works have highlighted the use of IoT supported by some remote processing. However, as IoT is a relatively recent paradigm, the oldest collected articles [20], [21] did not use the term IoT explicitly. Kartsakli et al. [20], for example, focuses on the use of communication protocols,

| Title | Year | Application Context | Main Architecture Components | Metrics | Resources Grouped by Location | Sensors Number of processing cores per machine | Mean response time per layer |
|-------|------|---------------------|-----------------------------|---------|-----------------------------|---------------------------------|-------------------------|
| [20]  | 2014 | Healthcare          | Sensor Network              | Throughput; percentage of successfully delivered packets; MRT, energy efficiency | No    | No                          | No                             | No                      |
| [21]  | 2015 | Healthcare          | Cloud                       | Waiting time                  | Yes   | No                          | No                             | No                      |
| [22]  | 2017 | Healthcare          | IoT and Cloud               | Number of Cyber Attacks       | No    | No                          | No                             | No                      |
| [23]  | 2017 | Healthcare          | IoT                         | Attack detection rate, throughput | No    | No                          | No                             | No                      |
| [24]  | 2018 | Healthcare          | Fog and Cloud               | MRT                           | Yes   | No                          | No                             | No                      |
| [25]  | 2019 | Healthcare          | IoT and Private DC          | MRT, drop rate, number of requests in the systems, utilization | Yes   | No                          | No                             | No                      |
| [26]  | 2020 | Without specific application context | IoT, Fog and Cloud | Utilization, MRT, throughput, and availability | Yes   | No                          | No                             | Yes                     |
| [9]   | 2020 | Healthcare          | IoT, Edge, Fog and Cloud    | Availability, Equipment Cost  | Yes   | No                          | No                             | No                      |
| Our Work | Healthcare | IoT, Edge, Fog and Cloud | MRT, utilization, system number of messages, throughput, drop rate | Yes   | Yes                         | Yes                            | Yes                     |
proposing the cloud-assisted RLNC-based MAC protocol (CLNC-MAC) and develop a mathematical model for the calculation of the key performance metrics. The term cloud in Kartsakli’s work refers to the cluster of sensors that communicate through a mesh network. They show the importance of central coordination in fully exploiting the gain of RLNC under error-prone channels. Marcu et al. [21] present an e-health solution based on healthcare information systems supported by hybrid (public-private) cloud computing with a focus on the medical imaging department. The other works used IoT devices as a component of the architecture and, in most cases, also used the fog layer. Our proposal explores not only IoT but also the edge, fog, and cloud layers as complementary resources. Only Santos et al. [9] explored such components. Thus, Santos et al. [9] propose surrogate models to estimate the availability of e-health monitoring systems, applying a multi-objective optimization algorithm to improve system availability considering component costs as a constraint. We believe that it is essential to explore all four layers to decrease the coupling between components and improve system performance. Santos et al. [9] did not explore performance issues and did not offer the same architectural contributions that we highlight in the comparative table.

B. METRICS

The more evaluation metrics, the better the understanding of the system behavior. Our study covers the largest number of performance metrics: MRT, utilization, system number of messages, throughput, and drop rate. Besides, we observed usage and MRT at different points in the model, not just in general. In contrast, two studies [22], [23] investigated metrics related to information security. Strielkina et al. [22] simulated attacks on IoT vulnerabilities and how to recover from them. Lomotey et al. [23] have proved that distributed health information system threats such as the denial of service, man-in-the-middle, spoofing, and masking can be effectively detected through simulations with Petri nets. Two other works studied the metric of availability [9], [26]. Andrade et al. [26], for example, revealed that adopting a Fog device can improve availability. However, the performance is only improved in certain conditions, like when the environment is not full capacity. Both security and availability issues are critical. However, the area explored by our work needed a more comprehensive and detailed performance study.

C. RESOURCES CAPACITY ANALYSIS

Normally, the main objective of the performance evaluation of systems – whether by measurement, simulation, or modeling – is to predict the system’s future behavior and be prepared to meet requests demand. Another critical point is to avoid wasting computational resources [31]. Therefore, analyzing the resources capacity to support the generation of IoT data is essential, especially in time-critical systems, such as healthcare. Almost all related works varied the resource capacity and observed different scenarios. Only three studies did not observe this aspect [20], [22], [23]. Kartsakli et al. [20] did not observe the issue of capacity as it focused on communication protocols. Strielkina et al. [22], and Lomotey et al. [23] did not study capacity issues as they focused on information security.

The last three comparing criteria have added a significant contribution to the theme because they are unique characteristics of the model proposed in this study. Sensors Grouped by Location refers to how the proposed model represents different groups of sensors. The related work uses only a single group of sensors and thus assigns a single arrival rate. Our model allows to assign different arrival rates depending on their location. Such a location in our model is characterized as being a hospital room. This unique functionality aims to imitate reality, because depending on the location, the sensors may have different criticisms, requiring more or less frequent data sending. Our model also has the unique feature of representing Number of Processing Cores per Machine. Both the fog (physical) and the cloud (VM) have machines with multiple cores. The model allows varying the capacity of the number of machines and each machine’s capacity in terms of cores. It is observed that in the related works, the number of cores is somewhat abstracted, and only the number of machines is observed. We believe that this feature is essential to represent the reality of multi-layer architecture more accurately. Finally, our model also has the differential of calculating two different response times (Mean Response Time per Layer), one for the fog and one for the cloud. We believe it is an essential requirement because our model expresses two types of clients, the client (doctors) who stays inside the hospital and requires a speedy response time, and another one that does not have the same need (family members, for example). Thus, the calculated MRTs allow to have a deep understanding of each part of the system. These three features further reinforce the model’s contribution to the IoT area supported by remote processing.

III. SYSTEM ARCHITECTURE

A. OVERALL SYSTEM ARCHITECTURE

A typical system architecture of an IoHT infrastructure for medical monitoring in hospitals or medical centers is presented in Figure 1. The IoHT infrastructure in consideration is constituted of three main computing layers including (i) a cloud computing layer at cloud data centers for the remote access of distant clients (e.g., family members, overseas medical professionals), (ii) a fog computing layer at fog data centers for the local access of internal clients (e.g., medical doctors, medical physicians) and, (iii) edge computing layer at local medical treatment rooms for health sensor data aggregation and integration. At the very edge of the computing network, edge computing layer offers real-time health data monitoring and aggregation by using edge devices to periodically collect and process raw data of patients’ health in each room. To provide health monitoring services in a hospital or a medical center, edge computing layer is designed with multiple edge nodes to perform health data
collection and processing at every room on every floor of a building. In the middle layer, the fog computing layer is composed of (i) a number of fog nodes for parallel processing of data regardless its source, with (ii) an edge-fog gateway for data aggregation and load balancing between edge and fog computing layers, and with (iii) a fog-cloud gateway for data transactions between fog and cloud computing layers. At the very center of the whole IoHT infrastructure, cloud computing layer offers powerful computing and storage capabilities for advanced data processing. It also plays a role of a center hub for accesses of distant clients to common medical data.

B. LIFE-CYCLE OF MESSAGES

The architecture also indicates the life cycle of data packages and operational behaviors of the system for healthcare monitoring purposes. Particularly, patients’ medical data is periodically collected by IoT health sensors, then automatically aggregated and encapsulated by edge devices (e.g., embedded system boards, IoT edge gateways) into medical messages. These medical messages are then transmitted to an internal fog center through an edge-fog gateway. The edge-fog gateway plays a role as an entrance door for data distribution and load-balancing to computing fog nodes. For the sake of high processing performance at local medical centers, we assume to use bare physical machines as fog nodes in the fog layer. Medical messages are processed on fog nodes by specialized medical services and applications, which are customized for the types of medical data as well as for the local medical departments. The processed medical data in the fog layer is delivered to internal clients’ front-end interface directly and also transmitted to remote cloud data centers for further data processing and storage through a fog-cloud gateway. The cloud layer offers a high level of processing and storage with multiple virtual machines (VMs) to deliver the processed data to distant clients. It is then a matter of curiosity that, (i) how the variation of periodic interval for medical data generated by health sensors attached on patients’ body and collected by edge devices in edge computing layer impacts the performance metrics of provided services (e.g., mean response time, throughput) and, (ii) how a specific configuration of each layer in the IoHT infrastructure and its variation impact the performance metrics of data transactions from patients’ attached health sensors all the way to internal clients at local medical centers through fog computing centers and to external clients at distant areas through cloud computing centers.

C. ASSUMPTIONS AND DISCUSSIONS

For the simplification of modeling, several assumptions on the architecture and operations of the IoHT infrastructure in consideration are given as follows.

- **Edge layer**
  - [e1]: Heterogeneous types of medical sensors are not considered in modeling. We model the generation of medical data by all sensors attached to patient in a room as a data burst to an edge device of the room.
  - [e2]: Various types and latency of communication between medical sensors and edge devices are not taken into account in the modeling. In practice, the connection is formed by wireless
communication. But, we neglect the negative impact of near-distance communication in edge layer on the overall performance metrics.

- **[e3]**: Communication latency of the connection between edge and fog layers is simply assumed to be an edge-fog delay of the data transmission from each edge node to the fog layer, which will be taken into account in the modeling.
- **[e4]**: Periodic health check and medical data collection are independent to each other and thus the arrival of requests is exponentially distributed with an arrival rate $\lambda$.

**Fog layer**

- **[f1]**: Advanced load balancing in fog layer is not taken into consideration. Jobs received from edge layer through edge-fog gateway are equally distributed to each of the fog nodes in the fog layer. For the simplification in modeling, load balancing problem of fog nodes is not our main focus. A comprehensive extension in future work could consider this problem for fog computing layer based on round-robin mechanism [32].
- **[f2]**: Heterogeneous configuration of each fog node is not of our main focus. We assume that fog nodes and their processing capabilities and resources are identical while data processing on each fog node is independent to one another. Each fog node can have multi-core CPU for parallel processing. We neglect the role of data storage in fog layer.
- **[f3]**: Communication latency of the connection between fog and cloud layers is supposedly considered as a simple fog-cloud delay of data transmission from fog layer to cloud layer.

**Cloud layer**

- **[c1]**: Data processing in cloud layer is supposedly performed on its virtual machines (VMs) instead of its bare physical machines. Therefore, we assume to not take into account bare physical machines and associated storage devices in the modeling to reduce the modeling complexity since we mainly consider the components (i.e., virtual machines) in charge of processing medical data in real-time scenarios.
- **[c2]**: As of modern cloud computing systems, virtual machines in cloud layer can have multiple cores for parallel processing. And at a certain time, the cloud layer can elastically scale and balance access load of a certain number of distant clients on a multi-core virtual machine.

**IoHT Infrastructure**

- **[i1]**: Pure performance of data transactions between IoT sensors and internal/external clients (e.g., medical professionals, medical physicians, family members) is the main focus of the modeling, therefore the involvement of physical components and their operational availability are minimized. And that, failure and recovery behaviors of components are not considered in the modeling for performance evaluation [33].
- **[i2]**: Our main focus is (i) to explore the bottle-neck in the considered IoHT architecture in real-time medical data transmission, (ii) to explore the impact of configuration alteration in fog and cloud layers on the performance metrics and (iii) to realize the performance-related trade-offs between internal clients at local places and distant clients. Therefore, the complexity of system architecture of the overall IoHT infrastructure is diminished to simplify performance models using queuing network theory. In this way, we can put more efforts for performance evaluation of the system for real-time medical applications.

**IV. QUEUING MODEL**

Figure 2 illustrates a model based on queuing theory for the presented architecture. All the model components follow the same name patterns used in the architecture description. All the aforementioned assumptions were attended in the model. Java Modeling Tools (JMT) [34] tool was used to model and evaluate the proposed scenario. JMT is a kit of open-source tools to simulate and evaluate communication systems’ performance based on queue theory [35].

Data flow occurs from left to right. Sensors generate requests within a predefined time interval following a particular probabilistic distribution. The model has multiple entry points, corresponding to the hospital rooms. Each room generates constant data collected from patients in possibly distinct arrival rates. Each room has an edge device which works as a gateway between the edge and the fog layer. Such a device is represented by a queue and a unique internal server (M/M/1 service station). The rooms have n patients, depending on the hospital organization. The arrival rate will depend on the number of patients and the distribution of data generation performed by the patients’ sensors. We consider that the rooms are organized in floors. Each floor has some distance to the fog layer. This way, there is a delay (“delay edge-fog”) from each floor to the fog layer. Delay components do not have a specific service; that is, it is only a component that causes a delay in the transmission of a request, simulating a delay of the network.

In the fog layer there are two gateways, one as entry point and other as exit point. When arriving at each gateway, these messages can be distributed following a specific load balancing strategy. We consider in this work the equally distribution strategy. Since the random distribution is not time consuming, the time to perform such distribution is unconsidered in this paper. Cloud and fog have a service time referring to the time to save the data and forward it to other analysis services for alert notices that can be applied. It is worth mentioning that the cloud is usually more potent than the fog, but the fog is closer to the sensors. It is assumed that health data received in each element of the general system will be processed.
considering a Fist-In-First-Out (FIFO) queuing discipline. The queues from the fog and the cloud are modeled considering the M/M/c/K queue model. The main parameters of such stations are: the queue size, service time and number of internal servers (which we consider as processing cores in this work). The M/M/c/K model refers to c service stations. Each service station has a capacity limitation of K nodes. The fog has a sink station corresponding to the place where the doctors can access sensitive and real-time data. Between the fog and the cloud there is the delay fog-cloud. Particularly, such a delay takes the longest time to finish due to the significant geographic distance of the cloud service. In the cloud, multiple Virtual Machines can be used, however, in this paper we simulated the use of only one powerful VM, since public clouds today offer VMs with incredible high number of cores (72 vCPU cores for example in the Amazon AWS). Finally, in the cloud layer there is a sink where the cloud processed data can be accessed.

V. SIMULATION RESULTS

This section presents Simulation analyzes using the proposed model. Table 2 shows the input parameters used for each model component. The X tag indicates that the component does not have a capacity definition for the queue. The time column represents the service time for the queue components. The time between for the delay components represents communication time.

Next subsections present three use case scenarios (A and B). The scenario A investigates the variation of fog nodes. The scenario B explores the variation of cloud VM cores.

2Amazon AWS VMs: https://aws.amazon.com/pt/ec2-instance-types/

TABLE 2. Simulation parameters.

| Component type | Component | Time (ms) | Queue Size |
|----------------|-----------|-----------|------------|
| Queue          | Edge Devices | 3         | 30         |
|                | Fog nodes   | 30        | 250        |
|                | Cloud VMs   | 50        | 5000       |

| Delay          | Edge-fog 1 | 6         | X          |
|                | Edge-fog 2 | 12        | X          |
|                | Fog-cloud  | 2000      | X          |

TABLE 3. Scenario A component parameters.

| Component  | Number of Machines | Number of Cores |
|------------|--------------------|-----------------|
| Edge       | 5                  | 1               |
| Fog        | 1/2/3/4            | 2               |
| Cloud      | 1                  | 4               |

A. SCENARIO A - FOG CAPACITY VARIATION

This section presents a simulation of the health monitoring system by varying the fog layer’s number of resources. Table 3 presents the configuration used in scenario A experiments. The number of fog nodes was varied in 1, 2, 3, and 4 nodes, maintaining the fixed values for the other components. Figure 3 presents the results considering different number of fog nodes.

Figure 3(a) shows the MRT referring to the system’s first exit point (Fog Client MRT). In general, maintaining the processing capacity, the greater the number of resources, the lower the response time. The MRT is much lower with 3 and 4 fog nodes than with 1 and 2 fog nodes. The configurations with 3 and 4 fog nodes obtained an MRT between 46 and 174ms. Configurations with 1 and 2 fog nodes peaked at over 3500ms. The stabilization of MRT growth for 1 and 2 fog nodes is due to the system having saturated all its resource capacity up to the Fog Client component. This saturation occurs from the arrival rate (AR) equal to 0.011msg/ms for 1 fog node and 0.015msg/ms for 2 fog nodes. The fog node’s
saturation can be justified in the fog utilization 3(c). For 1 and 2 fog nodes at the arrival rate points mentioned above, the fog utilization reaches 100% of the capacity. When such saturation and stabilization of MRT growth occurs, loss of messages begins (see drop rate in Figure 3(h)). The drop rate for 1 and 2 fog nodes start to grow exactly at the two inflection points: $AR = 0.011 \text{msg/ms}$ and $AR = 0.015 \text{msg/ms}$. Such growth is expected due to the depletion of resources. In this
section, we will give a hypothetical example of practical use of the results for each metric. Let us say that the system designer is not sure what the system’s arrival rate will be, but that it will vary between 0.01msg/ms and 0.024msg/ms. The system designer wants a Fog Client MRT below 200ms. In this case, the model suggests that the system designer can deploy 3 fog nodes sufficient to meet the demand.

Figure 3(b) shows the MRT considering the system’s second exit point (Cloud Client MRT). In this case, it is expected that the response time will be longer than the Fog Client because there are significant communication time and certain processing in the cloud as well. Thus, as expected, comparing the graph 3(a) and 3(b), there is a significant increase in the MRT in the cloud in all fog resources configurations.

The MRT for 2, 3 and 4 nodes are equivalent until AR = 0.016msg/ms in about 2200ms. The MRT for 1 node is in average 5800ms to all arrival rates. The highest variation occurs from the AR = 0.021 with 3 and 4 nodes, because much more messages reach the cloud and are not dropped. Now, let us say that the system designer is sure that the system arrival rate will be 0.012msg/ms and that the MRT in the Cloud Client should be below 10,000ms. In this case, the use of one fog node would meet the demand.

Figure 3(c) shows the fog layer utilization. There is a significant difference between the lines, considering the four fog capacities. However, there is only a full saturation of resources (100%) with 1 and 2 fog nodes. Such saturation happens at different times for 1 and 2 fog nodes. For 1 and 2 fog nodes, saturation occurs at AR = 0.011msg/ms and AR = 0.015msg/ms, respectively. The difference in usage depends on the workload. There is a small difference of 10% between 3 and 4 fog nodes observing the extremes with minimum AR (0.01msg/ms). There is a more significant difference of about 20%, considering the maximum arrival rate (0.024msg/ms). Let us say that the system designer only requires that the fog utilization being between 70% and 90% with an arrival rate of 0.019msg/ms. In this case, 3 fog nodes would be ideal to meet the imposed requirements.

Figure 3(d) shows the cloud utilization. The usage increases proportionally to the arrival rate for 3 and 4 fog nodes and mostly for 2. However, with 1 and 2 fog nodes, there is stagnation at 40% and 80%, respectively. With 3 and 4 fog nodes the utilization is very similar, reaching 100% with AR = 0.021msg/ms. Such stagnation is also caused by the bottleneck of resources in the fog node layer. This bottleneck makes it possible to discard requests beyond capacity, making fewer requests arrive at later architecture components. The available capacity of the cloud is wasted by a low performance at the Fog layer. Let us say that the system designer looks for an arrival rate between 0.012 and 0.016msg/ms and only requires that the cloud utilization being above 50% without data loss. In this case, at least 3 fog nodes would meet the requirement.

Figure 3(e) shows the edge utilization. There is a relatively linear increase proportional to the increase in arrival rate. The utilization level is shallow at all arrival rates, reaching just 16% at the highest arrival rate. The low utilization is due to two factors. First, the edge devices’ service time was configured with a shallow value (5ms) as it represents only the time for forwarding messages. Another factor is that each edge device receives data from a single data source, that is, from a single hospital room.

Figure 3(f) shows the number of messages within the system. Until AR = 0.019msg/ms, the system number of messages are similar to the four configurations. After that, the arrival rate becomes so high that for 3 and 4 fog nodes, the cloud queue reaches its maximum capacity (5,000msg). The lower configurations (1 and 2) do not reach the same maximum cloud queue capacity because in this case there are dropped messages.

Figure 3(g) shows the general throughput of the system. The higher the system’s processing capacity, the lower the drop rate and the higher the throughput. When there is no data loss, a system throughput will equal the total system arrival rate. The evaluated model has 6 hospital rooms, so for AR = 0.01msg/ms there is a result of 0.01 × 6 = 0.06msg/ms. The throughput is very low and constant for 1 fog node because, in the second arrival rate, the throughput stagnates at 0.06msg/ms. Therefore, with only 1 fog node, it would be possible to serve the 6 hospital rooms with a low arrival rate of around 0.01msg/ms. After this point, there would be data loss. For 2 fog nodes, the throughput stagnated at 0.14msg/ms because the resources were exhausted.

Figure 3(h) shows the system’s drop rate. As expected, there are no discarding requests for 3 and 4 fog nodes in the system. For 1 and 2 fog nodes, there is an increasing number of discards, starting at the arrival rate of 0.01msg/ms and 0.018msg/ms, respectively. The drop rate increase is caused by the depletion of resources in 1 and 2 fog nodes and is reflected in several system metrics. Let us say that the system designer allows data to be lost but that such loss does not exceed a rate of 0.05msg/ms. In this case, 2 fog nodes would be sufficient to meet the requirement.

### B. SCENARIO B - CLOUD CAPACITY VARIATION

This section shows the model simulation results by varying the cloud’s processing capacity in terms of the number of virtual machines. Table 4 presents the configuration used in scenario A experiments. The variation was performed with 1, 2, 3, and 4 virtual machine cores. Figure 4 presents the corresponding results.

Table 4. Scenario B component parameters.

| Component | Number of Machines | Number of Cores |
|-----------|--------------------|-----------------|
| Edge      | 6                  | 1               |
| Fog       | 3                  | 2               |
| Cloud     | 1                  | 1/2/3/4         |

Figure 4(a) shows the MRT considering the Fog Client output. As the capacity variation was performed in the cloud, there was no distinction between MRT’s behavior for these different cloud capacities. In other words, the Fog Client continues to respond within the 200ms requirement for all configurations.
FIGURE 4. Results of model simulations considering distinct number of cloud VM cores.

MRT does not change as the measurement of this MRT occurs in a component before the cloud component. However, for the four capacities together, significant growth in MRT is observed, reaching a time of 180ms. Still, the Fog Client MRT is much lower than most Cloud Client MRT, as can be seen in Figure 4(b).
Figure 4(b) shows the Cloud Client MRT. Unlike the Fog Client MRT, here, there is a significant distinction between the four cloud capacities. The greater the capacity of the cloud, the lower the MRT obtained. The lowest results for 2, 3, and 4 cloud VM cores ranged between 2000 and 2300ms. A higher capacity and a lower arrival rate result in an MRT very dependent on the communication time between the fog and the cloud, configured with the time of 2000ms. Despite this initial stability, for capacities 2, 3 and 4 there is a sudden rise in MRT at points AR = 0.012msg/ms, AR = 0.016msg/ns, and AR = 0.021msg/ms. This increase is again due to the high use of resources that reached 100% precisely at these points (see Figure 4(d)). Considering 1 cloud VM core, with constant 100% utilization, the MRT has reached an exorbitant 250 seconds. Let us say the system designer is looking for an arrival rate of about 0.016msg/ms. The simulation would allow the system designer to identify that 3 VM cores would obtain an MRT of 84800ms and 4 VM cores would obtain an MRT of approximately 2154ms. Therefore, in this case, the designer would undoubtedly prefer to rent 4 VM cores instead of 3.

Figures 4(c) and 4(d) show the fog and cloud utilization, respectively. Similar to the behavior of the Fog Client MRT, the fog utilization does not suffer any change due to the cloud capacity variation. However, there is a strong relationship between the use of fog and the increasing workload. In the cloud, as previously mentioned, the utilization reaches 100% in all scenarios, and the lower the capacity, the earlier this top is reached. For 1 cloud VM core, the maximum capacity is obtained from the first arrival rate. For 2, 3 and 4 cloud VM cores, depletion occurs at AR = 0.012msg/ms, AR = 0.016msg/ms and AR = 0.021msg/ms. Figure 4(e) shows the edge utilization. The result is identical to the variation in the fog capacity (scenario A), as both varied components occur after the edge layer. There is a relatively linear increase proportional to the arrival rate. However, it is essential to note that the level of utilization is low at all points, reaching 16% with no higher arrival rate.

Figure 4(f) shows the number of messages within the system. The behavior of capacities 2, 3, and 4 are similar, distinguished by the moment of the rise in the number of messages. The depletion of cloud resources causes such a rise. At the bottom of the graph, the system’s number of messages without data loss varies between 66 and 180 messages. However, with the depletion of resources, the number of messages is approximately 5,000, which is equivalent to the queue’s size in the cloud.

Figure 4(g) shows the overall system throughput. The greater the number of resources, the greater the throughput will be. The throughput for 2, 3, and 4 cloud VM cores are the same in the first three arrival rates. This equality occurs because there is no data loss in these three points, as can be seen in the drop rate graph (Figure 4(h)). At the point of maximum arrival rate (0.024msg/ms), the throughputs are almost equidistant: 1 VM core = 0.1101msg/ms, 2 VM cores = 0.1297msg/ms, 3 VM cores = 0.1493msg/ms, and 4 VM cores = 0.1691msg/ms. This difference of 0.01msg/ms is equivalent to 360,000 messages per hour. Therefore, the system designer must carefully observe the throughput metric before deciding how many VM cores to configure to meet the demand.

Figure 4(h) shows the system’s drop rate. As mentioned earlier, the smaller the number of resources, the greater the likelihood of data loss. With 4 VM cores, there was no data loss in any of the simulated arrival rates. For 1 VM core, there is loss in all simulated scenarios. For 2 and 3 VM cores, data losses started from points AR = 0.013msg/ms and AR = 0.018msg/ns, moments in which resources were exhausted.

VI. CONCLUSION AND FUTURE WORKS

This work proposed a queuing model of type M/M/c/K to represent and evaluate the performance of a health monitoring scenario corresponding to a hospital or medical center in the context of IoHT. The evaluated architecture includes IoT sensors, edge, fog and cloud components. The model allows estimating a significant number of metrics, such as: mean response time, the utilization level, and drop rate. The model offers unique contributions compared to related work. The model is composed of four layers, more significant number than the literature. The sensors are grouped by location, which permits configure distinct arrival rates based on sensors group. The model represents not only number of machines as capacity but also its internal number of processing cores. Finally, the model also enable us to calculated more than one mean response time, on the fog and on the cloud. Two scenarios were explored in simulation to exemplify the model utility. The first scenario observed the impact of varying fog resources under the system performance. The second scenario did the same but focusing on the cloud capacity. The simulations allowed to observe the relationship between the arrival rate and cloud/fog capacity variation under distinct perspectives. Moreover, the analysis indicated that even the fog being closer to sensors, cloud still plays an important role in the process. The results also show that queue size parameter has a significant impact on performance, and dropped messages can be avoided making small calibrations in fog/cloud resource capacities. As future work, it is intended to extend the model to explore different communication types between the components. More elements can also be included, such multi-clouds (e.g., public, private and hybrid). The type of the queues can be changed by including or excluding restrictions and buffer policies. Besides, further simulations can also be done, for example, exploring the load balancing strategies in the gateways.

REFERENCES

[1] B. Ç. Uslu, E. Okay, and E. Dursun, “Analysis of factors affecting IoT-based smart hospital design,” J. Cloud Comput., vol. 9, no. 1, pp. 1–23, Dec. 2020.
[2] D. Y. Huang, N. Athpore, F. Li, G. Acar, and N. Feamster, “IoTinspectorg: Crowdsourcing labeled network traffic from smart home devices at scale,” Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol., vol. 4, no. 2, pp. 1–21, Jun. 2020.

[3] M. Kaur, J. Malhotra, and P. D. Kaur, “A VANET-IoT based accident detection and management system for the emergency rescue services in a smart city,” in Proc. 8th Int. Conf. Rel. Infocom Technol. Optim. Trends. Future Directions (ICRITO), Jun. 2020, pp. 964–968.

[4] M. Ndiaye, S. S. Oyewobi, A. M. Abu-Mahfouz, G. P.Hancke, A. M. Kurien, and K. Djouani, “IoT in the wake of COVID-19: A survey on contributions, challenges and evolution,” IEEE Access, vol. 8, pp. 186821–186839, 2020.

[5] R. P. Singh, M. Javadi, A. Haleem, and R. Suman, “Internet of Things (IoT) applications to fight against COVID-19 pandemic,” Diabetes Metabolic Syndrome, Clin. Res. Rev., vol. 14, no. 4, pp. 521–524, 2020.

[6] Y. Dong and D.-Y. Yao, “IoT platform for COVID-19 prevention and control: A survey,” 2020, arXiv:2010.08056. [Online]. Available: http://arxiv.org/abs/2010.08056.

[7] M. B. Yassein, I. Hmeidi, M. Al-Harbi, L. Mrayan, W. Mardini, and Y. Kharnayesh, “IoT-based healthcare systems: A survey,” in Proc. 2nd Int. Conf. Data Sci., E-Learn. Inf. Syst., 2019, pp. 1–9.

[8] F. A. Silva, I. Fé, and G. Gonçalves, “Stochastic models for performance and cost analysis of a hybrid cloud and fog architecture,” J. Supercomput., vol. 77, no. 2, pp. 1537–1561, Feb. 2021.

[9] G. L. Santos, D. Gomes, J. Kelner, D. Sadok, F. A. Silva, T. P. Endo, and T. Lynn, “The Internet of Things for healthcare: Optimising E-health system availability in the fog and cloud,” Int. J. Comput. Sci. Eng., vol. 21, no. 4, pp. 615–628, 2020.

[10] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, “Fog computing and its role in the Internet of Things,” in Proc. 1st Ed. MCC Workshop Mobile Cloud Comput. (MCC), 2012, pp. 13–16.

[11] D. Carvalho, L. Rodrigues, P. T. Endo, S. Kosta, and F. A. Silva, “Mobile edge computing performance evaluation using stochastic Petri nets,” in Proc. IEEE Symp. Comput. Commun. (ISCC), Jul. 2020, pp. 1–6.

[12] T. A. Nguyen, D. Min, and E. Choi, “Stochastic reward net-based modeling and analysis of availability evaluation for cloud data center networks using stochastic reward nets,” J. Netw. Syst. Manag., vol. 29, no. 1, pp. 1–27, Jan. 2021.

[13] T. A. Nguyen, D. Min, and E. Choi, “A comprehensive evaluation of availability and operational cost for a software-defined network infrastructure,” J. Supercomput., vol. 74, no. 1, pp. 222–276, Jan. 2018, doi: 10.1007/s11227-017-2127-2.

[14] T. A. Nguyen, K. Han, D. Min, E. Choi, T. D. Thang, and Y.-J. Choi, “A stochastic reward net-based assessment of reliability, availability and operational cost for a software-defined network infrastructure,” J. Supercomput., vol. 75, no. 8, pp. 4657–4683, Aug. 2019, doi: 10.1007/s11227-018-2677-y.

[15] K. Satyanarayana, K. Sudha, N. T. Rao, and M. Chen, “Analysis of queueing model-based cloud data centers,” in Smart Technologies in Data Science and Communication. Singapore: Springer, 2020, pp. 251–261.

[16] F. J. Suárez, P. Nuño, J. C. Granda, and D. F. García, “Computer networks performance modeling and simulation,” in Modeling and Simulation of Computer Networks and Systems, Amsterdam, The Netherlands: Elsevier, 2015, pp. 187–223.

[17] T. Raz, “The art of computer systems performance analysis: Techniques for experimental design, measurement, simulation, and modeling (Raj Jain),” SIAM Rev., vol. 43, no. 3, pp. 518–519, Sep. 1992, doi: 10.1137/1021015.

[18] C. Pedroso and K. Fonseca, “Modeling weight rounding robin packet scheduler with Petri nets,” in Proc. 8th Int. Conf. Comput. Commun. Syst. (ICCS), vol. 1, Nov. 2020, pp. 342–345, doi: 10.1109/ICCS5002.2012.1182494.

[19] T. A. Nguyen, D. Min, and E. Choi, “A hierarchical modeling and analysis framework for availability and security quantification of IoT infrastructures,” Electronics, vol. 9, no. 1, p. 155, Jan. 2020, doi: 10.3390/electronics9010155.

[20] M. Bertoli, G. Casale, and G. Serazzi, “JiMT: Performance engineering tools for system modeling,” ACM SIGMETRICS Perform. Eval. Rev., vol. 36, no. 4, pp. 10–15, Mar. 2009.

[21] G. S. Fishman, Discrete-Event Simulation: Modeling, Programming, and Analysis, P. Glynn and S. M. Robinson, Eds. Berlin, Germany: Springer-Verlag, 2001, pp. 218–294 and 524, doi: 10.1007/978-1-4757-3552-9.

---

**FRANCISCO AIRTON SILVA** received the B.S. degree in information systems from the Instituto Federal do Piauí (IFPI), in 2008, and the M.S. and Ph.D. degrees in computer science from the Universidade Federal de Pernambuco (UFPE), in 2013 and 2017, respectively. He is currently a Professor of Information Systems with the Universidade Federal do Piauí (UFPI). In 2015, he studied at the Sapienza Università di Roma. He was a collaborative researcher in multinational technology companies, such as EMC and OKI. His research interests include distributed systems with emphasis on cloud-fog-edge integration, mobile computing, and systems performance evaluation techniques.
TUAN ANH NGUYEN received the B.Eng. and M.Sc. degrees in mechatronics from the Hanoi University of Science and Technology (HUST), Hanoi, Vietnam, in 2008 and 2010, respectively, and the Ph.D. degree in computer science and system engineering from the Department of Computer Engineering, Korea Aerospace University, Seoul, South Korea, in 2015. He was a Research Engineer with the TRON Laboratory, FPT Software, FPT Cooperation, Hanoi, and the F-Space Laboratory, FPT Technology Research Institute (FTRI), FPT University, FPT Cooperation, in 2008 and from 2009 to 2010, respectively. He was a Ph.D. Research Associate with the Network Security and Systems Laboratory (NS Laboratory), Korea Aerospace University (KAU), from 2011 to 2015. He was a Postdoctoral Research Associate with the Distributed Multimedia Systems Laboratory (DMS Laboratory), Konkuk University, Seoul, South Korea, from August 2015 to February 2016. He worked for Office of Research, University-Industry Cooperation Foundation, Konkuk University, from March 2016 to February 2020 as a (Research) Assistant Professor. He is currently a (Research) Assistant Professor with the Konkuk Aerospace Design-Airworthiness Research Institute (KADA), Konkuk University. His research interests include intelligent, dependable and secure digital twins, digital twin for urban aerial mobility, reinforcement learning based intelligent control for unmanned vehicles and robotic systems, generative adversarial networks (GANs) for digital twins, dependability and security of systems and networks, fault tolerance of embedded systems in aerospace and mechatronics, disaster tolerance and recovery of computing systems, integration of cloud/fog/edge computing paradigms, dependability and security analytical quantification for Internet of Things, cloud data centers, unmanned vehicles, mechatronic production chains, and e-logistics.

IURE Fé received the degree in computer science from the Federal University of Piauí, and the master’s degree in computer science from the Federal University of Pernambuco. He is currently working as a Systems Analyst with the Brazilian Army. His current research interests include distributed systems and performance evaluation of systems.

CARLOS BRITO is currently a Senior College Student with the Information Systems course, Federal University of Piauí. He has experience in computer science and information systems, with emphasis on systems and distributed systems evaluation. He is interested in researching themes related to distributed systems, including the Internet of Things and cloud computing.

DUGKI MIN received the B.S. degree in industrial engineering from Korea University, in 1986, and the M.S. and Ph.D. degrees in computer science from Michigan State University, in 1991 and 1995, respectively. He is currently a Professor with the Department of Computer Science and Engineering, Konkuk University. His research interests include cloud computing, distributed and parallel processing, big data processing, intelligent processing, software architecture, and modeling and simulation.

JAE-WOO LEE received the B.S. and M.S. degrees in aerospace engineering from Seoul National University, Seoul, South Korea, and the Ph.D. degree from the Department of Aerospace Engineering, Virginia Tech, USA, in 1991. He is currently a Professor and the Director with the Konkuk Aerospace Design-Airworthiness Research Institute (KADA), Konkuk University. He is also the President of the Korean Society of Design Optimization (KSDO) and also serves as the vice president for several academic societies, including the Korean Society of Aeronautics and Space Science (KSASS), and KOSSE. He is the corresponding author or the coauthor of over 570 publications, including 13 patents and 74 international journal articles. His research interests include multidisciplinary design and optimization, MDO, aerodynamic design and optimization, aerospace vehicle design for aircraft, space launcher, and UAV/Drones. He received several honors and awards in his academic career. He has been serving as the conference chair, the technical program chair, and the symposium chair for various international conferences, including APISAT, KSAS, and KSAA. He has served as a Specialized Member for Defense Acquisition Committee, Ministry of National Defense, South Korea, and for Policy Planning Committee/Business Committee, DAPA, MND, South Korea.

**VOLUME 9, 2021**

IEEE Access