Availability Assessment of Internet of Medical Things Architecture using Private Cloud

Thiago Valentim[1], **Gustavo Callou**[2], **Alison Vinicius**[2], **Cleunio França**[1], **Eduardo Tavares**[1]

[1]Centro de Informática,
Universidade Federal de Pernambuco, Recife, Brazil

[2]Departamento de Computação,
Universidade Federal Rural de Pernambuco, Recife, Brazil.

{tvb,cbff,eagt}@cin.ufpe.br, {alison.gsilva,gustavo.callou}@ufrpe.br

Abstract. *Investments in smart health applications are expected to rise to US$ 960 billion by 2030, and Internet of Things (IoT) have a prominent role in implementing such applications. For instance, hospitals have adopted IoT to collect and transmit patient data to health professionals, as critical patients must be monitored uninterruptedly. Therefore, health systems commonly require high availability, but availability assessment of health systems' architecture is not a common approach. This paper presents a modeling approach based on generalized stochastic Petri nets (GSPN) to evaluate the availability of Internet of Medical Things (IoMT) architecture based on a private cloud. A case study is adopted to demonstrate the feasibility of the proposed approach.*

1. Introduction

Internet of Things (IoT) is a technological paradigm promoting advances in several aspects of our daily lives, such as predicting natural disasters, autonomous vehicles, traffic monitoring, and smart hospitals [Wamba et al. 2013].

Particularly, global investment in IoT-based medical systems was estimated at US$217.34 billion in 2022, and it is expected to be around US$960.2 billion by 2030. The huge investment is also related to acquiring sensors and computing devices for remote patient monitoring (RPM) [Healthcare 2022]. Consequently, the term Internet of Medical Things (IoMT) has been coined to highlight the importance of IoT in medical applications.

IoMT systems may be considered critical because system failures may affect patient lives. Therefore, over the years, research has been carried out to conceive techniques to improve availability in IoTM applications [Tang and Xie 2021]. Private clouds are very important, as patient data are sensitive and must be securely stored by health units. However, the influence of the components of a private cloud is usually neglected.

Availability evaluation may contemplate several system components, which may have a distinct influence on system operation. In this context, stochastic models are prominent, as different system elements and architectures can be evaluated before implementing the real system.

This paper presents a modeling approach based on generalized stochastic Petri nets (GSPN) to evaluate the availability of Internet of Medical Things (IoMT) systems.

Our proposed GSPN model assumes an IoMT system in a private cloud, considering software and hardware components that compose the architecture (e.g., virtual machine, microcontroller and sensor). A case study is adopted to demonstrate the feasibility of the proposed approach, in which redundant components are added from a baseline architecture to improve system availability.

The remainder of this paper is organized as follows. Section 2 details related work, and Section 3 presents prominent concepts for a better understanding of this work. Section 4 presents the IoMT architecture, and Section 5 details the availability model. Section 6 presents experimental results, and Section 7 concludes this work.

2. Background

This section introduces important concepts for a better understanding of this work.

2.1. IoT architecture

Internet of Things (IoT) allows environmental components to be remotely monitored using existing network infrastructures, creating a prominent integration of distinct computer systems. Such integration results in efficient data gathering, monitoring and processing [Gokhale et al. 2018].

A basic IoT architecture [Jara et al. 2009] is divided into four layers (Figure 1): devices, communication, processing and presentation. Devices perform data gathering and contemplate, for instance, sensors and microcontrollers. The communication layer carries out data transfer using standard protocols, such as LoRA and ZigBee, for further processing. The processing layer manipulates data to execute system services, and the presentation layer provides mechanisms for end-user interaction.

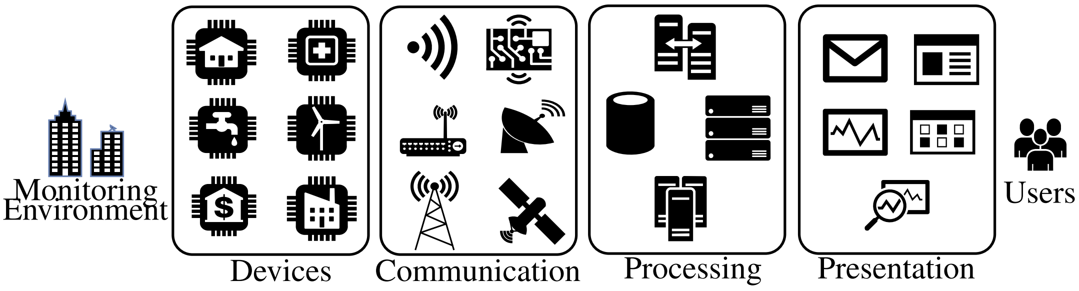

Figure 1. IoT Basic Architecture

2.2. Availability

IoMT systems usually deal with sensitive data, and, thus, high availability is an important attribute for those systems [Joyia et al. 2017]. For instance, an equipment failure cannot cause serious consequences for a patient (e.g., a monitoring device failure cannot lead to a false alert).

Availability is the probability of a system being in a functioning state. Steady-state availability (A) is commonly utilized, which takes into account the relationship between the system's mean time to failure ($MTTF$) and mean time to repair ($MTTR$):

$$A = \frac{MTTF}{MTTF + MTTR} \tag{1}$$

$$MTTF = \int_0^\infty R(t)dt \tag{2}$$

$$MTTR = \int_0^\infty 1 - M(t)dt \tag{3}$$

$M(t)$ is the cumulative distribution function representing the probability that a repair will occur within time t. $R(t)$ is the reliability function, which is the probability of a system performing its functions without failures for a period of time t.

2.3. Petri Nets

Petri nets (PN) are a graphical and mathematical modeling tool that can be adopted to represent several system types. For instance, parallelism, concurrency, asynchronous and non-deterministic activities are naturally expressed in a PN model [Murata 1989].

A Petri net is a bipartite directed graph in which places denote local states and transitions represent actions. Arcs connect places to transitions and vice-versa. Tokens may reside in places denoting a PN's state (i.e., marking). An inhibitor arc represents the unavailability of tokens in places, and the semantics of a PN is defined in terms of a token game (i.e., tokens are generated and consumed due to the firing of transitions).

This work adopts a specific PN extension, namely, generalized stochastic Petri nets (GSPN), which allows the addition of probabilistic delays to timed transitions or zero delays (and guard expressions) to immediate transitions (Figure 2).

The state space of a GSPN model can be translated into a continuous-time Markov chain (CTMC), and simulation techniques can also be adopted as an alternative to the generation of CTMCs [Maciel 2022].

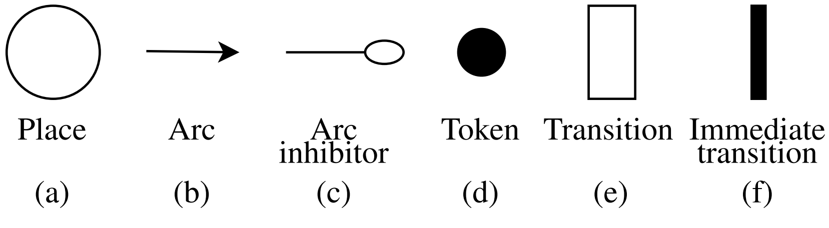

Figure 2. Basic elements of GSPNs

As an example, Figure 3 presents a model with a physical machine (left) and a virtual machine (right). A token in place *HostUp* (*VMUp*) indicates the physical device (VM) is operational, and a token in place *HostDown* (*VMDown*) denotes the device is unavailable. The firing of transition *tHostFail* (*tVMFail*) consumes a token from place *HostUp* (*VMUp*) and generates a token in place *HostDown* (*VMDown*), representing the device inoperability. Similarly, the firing of transition *tHostRepair* (*tVMRepair*) denotes the maintenance (recovery) of a device. In case of physical machine failure, the VM is also not operational. More specifically, immediate transition $t1$ is enabled due to the inhibitor arc, and the respective firing represents the VM failure (i.e., token generated in place $VMDown$).

As usually adopted in GSPN, operator # represents the number of tokens in a place (e.g. #HostUp), and $P\{exp\}$ indicates the probability of inner expression exp. These operators are utilized in Section 5.

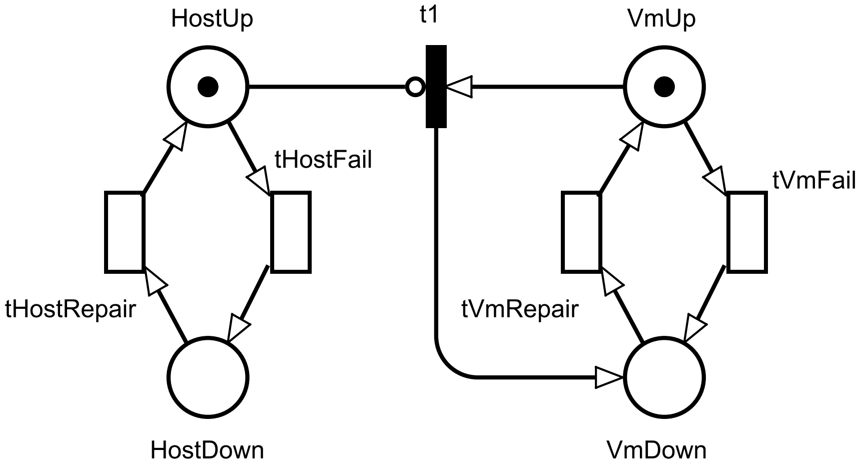

Figure 3. GSPN Example

3. Related work

Over the last few years, extensive research has been conducted to assess the performance of IoMT systems, and only some works have taken into account system failures or availability.

In [Macedo et al. 2014], the authors provide a method based on continuous-time Markov chains for assessing IoT infrastructures. The method contemplates passive and active redundancy, but results are not directly related to availability, and nothing is stated about IoMT. Rahmani et al. [Rahmani and Hosseini Mirmahaleh 2022] present a failure detection method for IoMT systems. The mechanism verifies operational servers and moves them to join the cluster to restore the system service. Despite the importance of such a work, a sensitivity analysis still needs to be carried out. Razdan et al. [Razdan and Sharma 2021] proposed an IoMT architecture based on cloud and fog computing. However, the architecture is not quantitatively evaluated and nothing is stated about availability.

In [Nguyen et al. 2021a], the authors proposed an approach to assess the impact of load-balancing techniques on the performance of a medical information system. That work adopts stochastic reward nets for system modeling. Santos et al. [Santos et al. 2020] studied a combination of stochastic models with a multi-objective optimization algorithm to analyze the influence of failures on an e-health system. Nevertheless, a sensitivity analysis is not carried out. Md Ashraf et al. [Uddin et al. 2018] presented a patient-centric IoMT architecture using blockchain technology. The authors use a system prototype instead of (formal) stochastic modeling.

In [Dilibal 2020], the authors describe an IoMT architecture based on edge computing for patient monitoring. Performance and availability evaluations are neglected.

Nguyen et al. [Nguyen et al. 2021b] proposed a hierarchical modeling approach for assessing IoMT infrastructures. The approach adopts fault trees and Markov chains, focusing on availability and security issues.

Unlike previous works, this paper proposes a modelling approach based on GSPN for evaluating Internet of Medical Things architecture using a private cloud. Our approach also identifies the system components with the greatest impact on system availability.

4. IoMT Architecture

An IoMT architecture defines a smart environment that contemplates electronic devices and sensors to monitor physiological signals from patients that may or may not be hospitalized [Ramson and Moni 2016].

The architecture should allow physicians to access and analyze patient data in real-time, making better-informed decisions about a patient care. This environment includes communication protocols, data storage, data analytics, visualization tools, and other hardware and software components to enable medical personnel to monitor and remotely manage patient health. Additionally, the architecture can collect data from various sources, such as wearable and mobile devices, providing a comprehensive view of patient status [Askar et al. 2022].

Figure 4 represents the IoMT architecture adopted in this work, which is based on [Vishnu et al. 2020] and has 5 layers. The sensor layer is composed of sensors responsible for collecting patients' physiological data. The data gathering layer is composed of devices that transmit data provided by sensors using the public network (i.e., Internet). The fog (computing) layer represents the intermediary server that immediately receives the data collected from patients. In this way, data are processed more quickly for urgent actions required to mitigate health problems [Nguyen et al. 2021a]. The private cloud layer is responsible for storing patient data for future assessment. A private cloud is required, as patient data are very sensitive and must be managed and securely stored by health units.

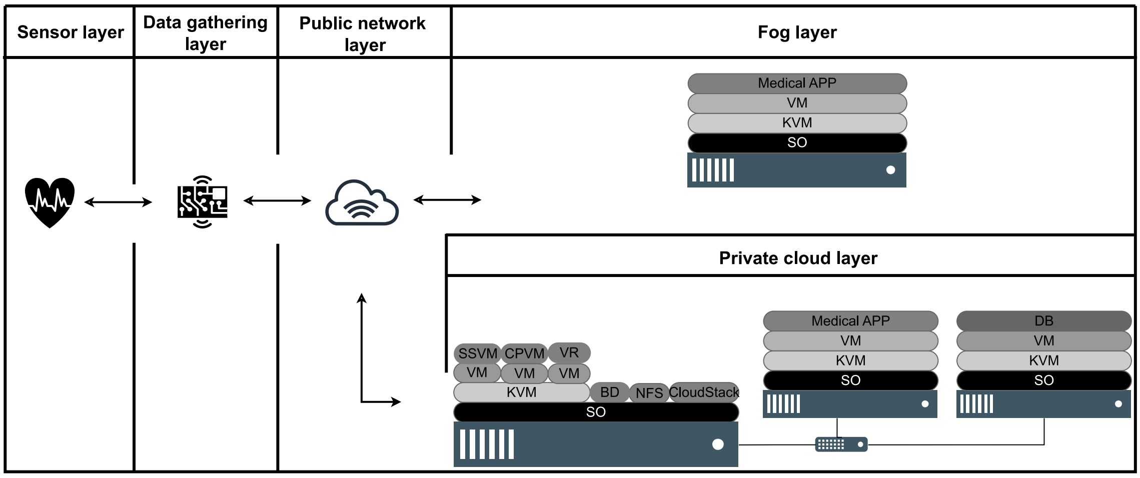

Figure 4. IoMT Basic Architecture

The architecture assumes the system is composed of 4 hosts at least. One host is deployed on the fog (for the processing application), and 3 hosts are adopted by the cloud computing infrastructure. Regarding the latter, one machine deals with the software platform (e.g., Cloud OpenStack [Sefraoui et al. 2012]), and two machines are related to data storage (i.e., medical storage software and DBMS).

5. Availability Model

This section presents the GSPN model (Figure 5) conceived for representing IoMT systems based on the adopted architecture (Section 4). The metric of interest is steady-state availability, and the system is only available if the 5 layers are operational. In other words, if a single layer is not working, the system is in a failure state. The public network is not explicitly represented, and the respective failure is assumed in the fog layer.

For a better understanding, the model is divided into three building blocks (i.e., submodels): cloud, fog and sensor. Additionally, we assume a system with four computers (hosts) and a single sensor for the sake of explanation. Later in this section, the addition of new components is explained. The sensor and data gathering layer is represented by the sensor block, such that a microcontroller is responsible for sending patient data to a health unit (i.e., a fog block). The fog layer has a host and the respective virtual machine (VM) running the medical application for data processing (APPMedicalFOG). The cloud layer adopts three machines for the cloud software platform (VMsSys), medical software for the storage (VMMedical) and the DBMS (VMDB).

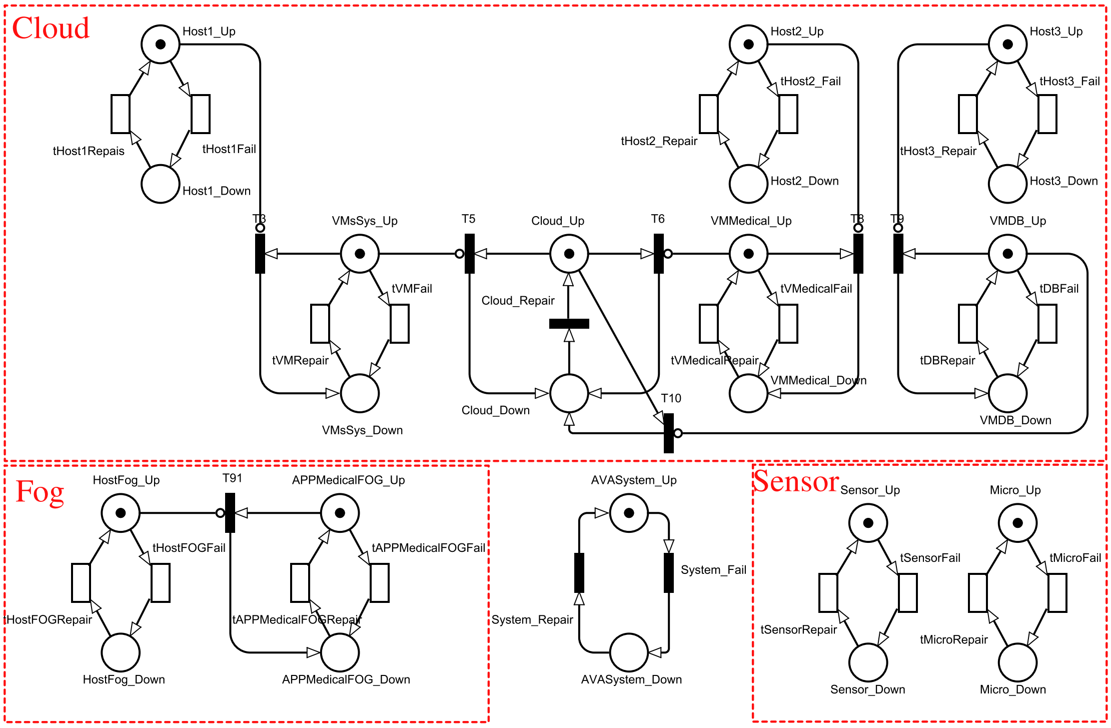

Figure 5. Availability Model

The sensor block is composed of model components, which indicate the device is operational (X_UP) or down (X_DOWN). Transitions *tXFail* and *tXRepair* represent the failure and maintenance of a device, and the respective delays are the device MTTF and MTTR. As previously explained, one sensor and one microcontroller are assumed.

The fog block adopts the GSPN model (Figure 3) described in Section 2.3, in which a virtual machine (medical application) is associated with a physical machine. In this case, the host failure ($HostFog\_Down$) also leads the application (APPMedicalFOG) to a failure state. The cloud block also associates each software component with a machine, and a host failure shuts down the associated VMs . Since the cloud layer requires the three hosts, immediate transitions $T5$, $T6$ and $T10$ may fire if a machine fails. In this case, the whole cloud is assumed nonoperational (token in place $Cloud\_Down$). The cloud maintenance is associated with transition $Cloud\_Repair$, and its guard expression (Table 1) demands the cloud software platform, DBMS and medical application working.

System availability is estimated using $P\{\#AVASystem\_Up = 1\}$. If one layer is not functioning, the system fails (token in place $AVASystem\_Down$). Transition $System\_Fail$ represents such a situation, and Table 1 depicts its guard expression. Similarly, the firing of transition $System\_Repair$ represents all layers are operational, which is also modeled using a guard expression (Table 1).

Table 1. Guard expressions

Transition	Guard Function
System_Fail	((#MICRO_Down>0) OR (#Sensor_Down>0) OR (#HostFog_Down>0) OR (#APPMedicalFog_Down>0) OR (#Cloud_Down>0))
System_Repair	((#MICRO_Up>0) AND (#Sensor_Up>0) AND (#HostFog_Up>0) AND (#APPMedicalFog_Up>0) AND (#Cloud_Up>0))
Cloud_Repair	((#VMsSys_Up>0) AND (#VMDB_Up>0) AND (#VMMedical_Up>0))

5.1. Spare components

Spare components can be represented by adding tokens in a place X_UP. For instance, two tokens in place $Host1\_Up$ ($\#Host1\_Up = 2$) indicate two machines: the primary and a spare. Besides, transitions need to adopt infinite-server semantics, which is adopted to represent parallel activities. In this case, the firing rate of a transition is linearly increased according to its enabling degree. The reader is referred to [Balbo 2001] for more details.

6. Experimental results

This section presents experimental results to demonstrate the feasibility of our technique. The focus is on the availability assessment of the adopted architecture taking into account redundant hosts.

Table 2. Values for Timed Transitions

Component	MTTF (h)	MTTR (h)
Host	1259.0	0.725
VM	2880.0	0.170
Microcontroller	44987.0	5.000
Sensor	28011.0	5.000

In this work, a design of experiments (DoE) [Montgomery and Runger 2010] is utilized with a l^k factorial design, in which the adopted factors ($k = 4$) are as follows: (i) database server (DB); (ii) cloud medical application (App) ; (iii) fog layer (Fog); and sensor layer (Sensor). For all factors, the levels (l) are 1 (primary component) and 2 (primary and a spare component). For each treatment (i.e., combination of factor levels), a model based on Figure 5 is created, and a stationary analysis is carried out to estimate steady-state availability.

Table 3. Treatments and results - availability

Treatments	BD	App	Fog	Sensor	Availability
1	1	1	1	1	0.996637186
2	2	1	1	1	0.997404010
3	1	2	1	1	0.997404014
4	2	2	1	1	0.998171416
5	1	1	2	1	0.997404011
6	2	1	2	1	0.998171409
7	1	2	2	1	0.998171412
8	2	2	2	1	0.999096690
9	1	1	1	2	0.996925795
10	2	1	1	2	0.997692839
11	1	2	1	2	0.997692837
12	2	2	1	2	0.998416981
13	1	1	2	2	0.997692842
14	2	1	2	2	0.998163831
15	1	2	2	2	0.998460440
16	2	2	2	2	0.999217760

The evaluation has been performed using the Mercury tool [Silva et al. 2013]. Database server, cloud medical application, and fog layer consider the full host (physical machine and the respective VM), and the sensor layer assumes a pair of sensors and microcontroller. A redundant component (level 2) takes into account another full host or an additional sensor and microcontroller. Table 2 depicts the values adopted for $tXRepair$ and $tXFail$ transitions, which are based on [Tang et al. 2004], [Kim et al. 2009], [productreliability 2022].

Table 4. Rank of effects

Factor/Interaction	Effect
App	0.000817
Fog	0.000754
DB	0.000743
Sensor	0.000225
DB*Sensor	-0.000063
App*Fog	0.000061

Table 3 presents the system availability (A) for each treatment, and Table 4 depicts the rank of effects. Effect [Montgomery and Runger 2010] is the change in availability associated with a change in a factor level, and the rank is presented in descending order, assuming the absolute values of all effects. Cloud medical application (App) is the most important factor, as adding a spare component, availability has the greatest improvement (almost 0.08%). For instance, assuming a downtime in hours during one year ($D = [1-A] \times 8760$), treatment 3 (App with redundancy) has a downtime of 22.74 hours, which improves in 7 hours the system outage in relation to treatment 1 (no redundancy). Other factors also have a significant effect, and they are followed by the interaction of database and sensor (BD*Sensor) as well as medical application and the fog layer (App*Fog).

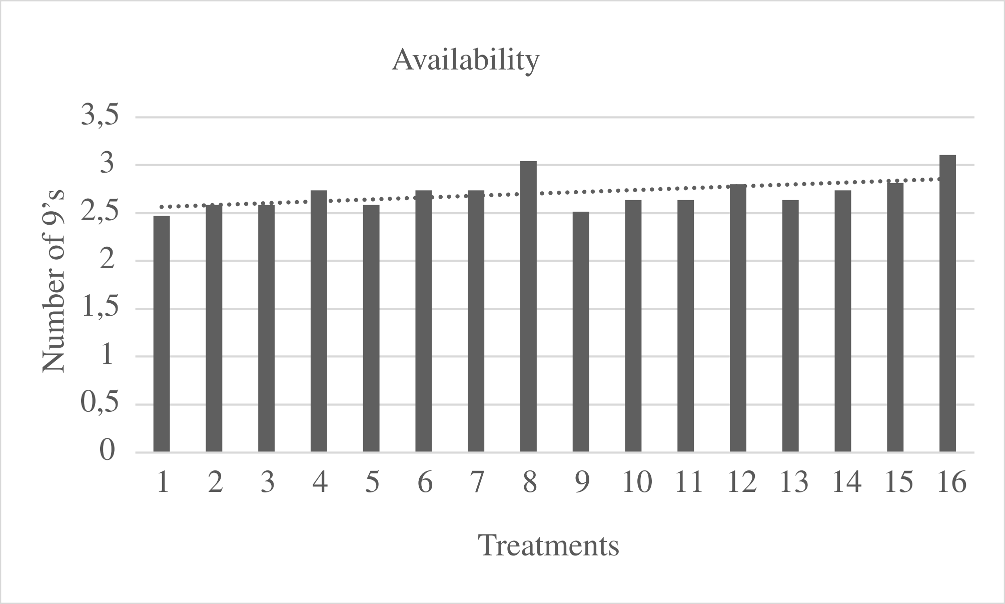

Figure 6. Availability results using the number of 9s

For a better visualization, Figure 6 depicts each availability (A) in Table 3 using number of nines ($-log_{10}[1 - A]$). The highest values are related to the adoption of spare components in all factors, but factor App has the greatest impact.

Results indicate distinct system configurations may be evaluated with the proposed approach concerning the availability of IoMT systems. The conceived technique is an additional tool for designers, which also allows to assess the influence of each component on system availability using effects.

7. Conclusion

This paper presented a modelling approach based on GSPN for assessing the availability of IoMT systems. The proposed model allows the evaluation of distinct system designs before modifying or implementing the real system or a prototype, which can be costly. Experimental results demonstrate the practical feasibility of the proposed approach. A sensitivity analysis was also considered to indicate components with the highest impact on system operation.

In future work, we plan to extend the proposed model to include performance and energy consumption assessment, such that the influence of system availability would also be jointly evaluated.

Acknowledgement

The authors would like to thank CNPq and FACEPE for the financial support for this work.

References

Askar, N. A., Habbal, A., Mohammed, A. H., Sajat, M. S., Yusupov, Z., and Kodirov, D. (2022). Architecture, protocols, and applications of the internet of medical things (iomt). *Journal of Communications*, 17(11).

Balbo, G. (2001). Introduction to stochastic petri nets. *Lectures on Formal Methods and PerformanceAnalysis: First EEF/Euro Summer School on Trends in Computer Science Bergen Dal, The Netherlands, July 3–7, 2000 Revised Lectures 1*, pages 84–155.

Dilibal, Ç. (2020). Development of edge-iomt computing architecture for smart healthcare monitoring platform. In *2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, pages 1–4. IEEE.

Gokhale, P., Bhat, O., and Bhat, S. (2018). Introduction to iot. *International Advanced Research Journal in Science, Engineering and Technology*, 5(1):41–44.

Healthcare (2022). Internet of things in healthcare market.

Jara, A. J., Zamora, M. A., and Skarmeta, A. F. (2009). An architecture for ambient assisted living and health environments. In *International Work-Conference on Artificial Neural Networks*, pages 882–889. Springer.

Joyia, G. J., Liaqat, R. M., Farooq, A., and Rehman, S. (2017). Internet of medical things (iomt): Applications, benefits and future challenges in healthcare domain. *J. Commun.*, 12(4):240–247.

Kim, D. S., Machida, F., and Trivedi, K. S. (2009). Availability modeling and analysis of a virtualized system. In *2009 15th IEEE Pacific Rim International Symposium on Dependable Computing*, pages 365–371. IEEE.

Macedo, D., Guedes, L. A., and Silva, I. (2014). A dependability evaluation for internet of things incorporating redundancy aspects. In *Proceedings of the 11th IEEE international conference on networking, sensing and control*, pages 417–422. IEEE.

Maciel, P. R. M. (2022). *Performance, reliability, and availability evaluation of computational systems, volume I: performance and background*. Chapman and Hall/CRC.

Montgomery, D. C. and Runger, G. C. (2010). *Applied statistics and probability for engineers*. John wiley & sons.

Murata, T. (1989). Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580.

Nguyen, T. A., Fe, I., Brito, C., Kaliappan, V. K., Choi, E., Min, D., Lee, J. W., and Silva, F. A. (2021a). Performability evaluation of load balancing and fail-over strategies for medical information systems with edge/fog computing using stochastic reward nets. *Sensors*, 21(18):6253.

Nguyen, T. A., Min, D., Choi, E., and Lee, J.-W. (2021b). Dependability and security quantification of an internet of medical things infrastructure based on cloud-fog-edge continuum for healthcare monitoring using hierarchical models. *IEEE Internet of Things Journal*, 8(21):15704–15748.

productreliability (2022). Tips for predicting product reliability.

Rahmani, A. M. and Hosseini Mirmahaleh, S. Y. (2022). Flexible-clustering based on application priority to improve iomt efficiency and dependability. *Sustainability*, 14(17):10666.

Ramson, S. and Moni, D. J. (2016). A case study on different wireless networking technologies for remote health care. *Intelligent Decision Technologies*, 10(4):353–364.

Razdan, S. and Sharma, S. (2021). Internet of medical things (iomt): overview, emerging technologies, and case studies. *IETE Technical Review*, pages 1–14.

Santos, G. L., Gomes, D., Kelner, J., Sadok, D., Silva, F. A., Endo, P. T., and Lynn, T. (2020). The internet of things for healthcare: Optimising e-health system availability in the fog and cloud. *International Journal of Computational Science and Engineering*, 21(4):615–628.

Sefraoui, O., Aissaoui, M., Eleuldj, M., et al. (2012). Openstack: toward an open-source solution for cloud computing. *International Journal of Computer Applications*, 55(3):38–42.

Silva, B., Callou, G., Tavares, E., Maciel, P., Figueiredo, J., Sousa, E., Araujo, C., Magnani, F., and Neves, F. (2013). Astro: An integrated environment for dependability and sustainability evaluation. *Sustainable computing: informatics and systems*, 3(1):1–17.

Tang, D., Kumar, D., Duvur, S., and Torbjornsen, O. (2004). Availability measurement and modeling for an application server. In *International Conference on Dependable Systems and Networks, 2004*, pages 669–678. IEEE.

Tang, S. and Xie, Y. (2021). Availability modeling and performance improving of a healthcare internet of things (iot) system. *IoT*, 2(2):310–325.

Uddin, M. A., Stranieri, A., Gondal, I., and Balasubramanian, V. (2018). Continuous patient monitoring with a patient centric agent: A block architecture. *IEEE Access*, 6:32700–32726.

Vishnu, S., Ramson, S. J., and Jegan, R. (2020). Internet of medical things (iomt)-an overview. In *2020 5th international conference on devices, circuits and systems (ICDCS)*, pages 101–104. IEEE.

Wamba, S. F., Anand, A., and Carter, L. (2013). A literature review of rfid-enabled healthcare applications and issues. *International Journal of Information Management*, 33(5):875–891.