Cluster Computing (2025) 28:275
https://doi.org/10.1007/s10586-024-04973-8

# Assessing the performance of a fault tolerant LoRaWAN architecture with a focus on the sensor layer and data retransmission strategy

Israel Araújo[1] · Vandirleya Barbosa[1] · Luiz Nelson Lima[1] · Luis Guilherme Silva[1] · Carlos Brito[1] · Iure Fé[1] · Lucas Lopes[1] · Ermeson Andrade[3] · Erico Leão[2] · Francisco Airton Silva[1]

Received: 27 April 2024 / Revised: 3 December 2024 / Accepted: 8 December 2024 / Published online: 25 February 2025


**Abstract**

Wireless Sensor Networks and LoRa are essential technologies for building low-power, long-range monitoring, and communication systems. Network and application servers are fundamental in providing sensor data in a LoRaWAN infrastructure. However, small changes in sensor configurations can lead to large variations in the performance of the architecture as a whole. The limited coverage of sensors and the lack of connectivity in remote areas make it difficult to obtain information, in addition, there needs to be an automated and low-cost system, as experiments in the real world require the cost of purchasing equipment. The proposed approach in this paper is to utilize stochastic Petri nets to analyze the performance of LoRaWAN networks with a focus on the sensor layer. This approach considers various factors, such as the data retransmission strategy, and evaluates its impact on the final throughput. Five case studies are presented, demonstrating the usefulness of the adopted methodology. The model is adaptable to other scenarios with characteristics similar to the LoRa architecture.

**Keywords** Sensors network · Lorawan · Performance analysis · Throughput · Petri Nets

## 1 Introduction

In recent years, several long-distance wireless communication technologies, such as LoRa (Long Range),[1] Sigfox,[2] Weightless[3] and Wi-Fi HaLow,[4] have gained prominence for their efficiency in scenarios that disable connectivity over large areas. Although some of these technologies, such as Sigfox, launched in 2010, and LoRa, developed in 2009, have been around for more than a decade, they continue to evolve. Currently, these solutions are

✉ Francisco Airton Silva
faps@ufpi.edu.br

Israel Araújo
israel.araujo@ufpi.edu.br

Vandirleya Barbosa
vandirleya.barbosa@ufpi.edu.br

Luiz Nelson Lima
luizznelson@ufpi.edu.br

Luis Guilherme Silva
luis.e@ufpi.edu.br

Carlos Brito
carlosvictor@ufpi.edu.br

Iure Fé
iuresf@ufpi.edu.br

Lucas Lopes
lucaslopes092020@ufpi.edu.br

Ermeson Andrade
ermeson.andrade@ufrpe.br

Erico Leão
ericoleao@ufpi.edu.br

1 Laboratory of Applied Research to Distributed Systems (PASID), Federal University of Piauí (UFPI), R. Cí-cero Duarte, no 905 - Junco, Picos, Piauí 64607-670, Brazil

2 Department of Computing, Federal University of Piauí (UFPI), Teresina, Piauí, Brazil

3 Department of Computing, Federal Rural University of Pernambuco, Recife, Pernambuco, Brazil

1 http://www.semtech.com/wireless-rf/lora.html.

2 https://www.sigfox.com/en.

3 http://www.weightless.org/

4 https://www.wi-fi.org/discover-wi-fi/wi-fi-certified-halow.

increasingly consolidated and widely used in modern Internet of Things (IoT) applications, due to their ability to provide robust, long-range and low-power communication. These wireless sensor networks support a wide range of applications, including infrastructure security [1], environmental monitoring [2], and sensing in smart cities [3]. Such applications aim to collect data on specific features or events. In many cases, the sensors are deployed in harsh and difficult-to-access environments, requiring them to operate for long periods without human intervention.

LoRaWAN, or wide area network, is an LPWAN (Low-Power Wide Area Networks) network technology that uses spread spectrum technology to enable data transmission over long distances[4]. LoRaWAN has high energy efficiency, wide coverage, and the ability to send data across obstacles. Furthermore, LoRaWAN can be used by companies and organizations without any restrictions, which makes it an affordable option for many applications. LoRaWAN was designed primarily to be used with sensors that must exchange information with a server at a low transmission rate and relatively long time intervals, where a transmission can occur every hour or even every few days [5]. Even with this configuration of long intervals between the generation of new data, the high dispersion and the large number of sensors can make it difficult to predict system performance, especially if the model used is not sufficiently accurate.

In a LoRaWAN architecture, many factors can influence the data flow generated at the sensor layer until it reaches the application layer. The network server and application server can easily become bottlenecks in data processing. In a previous study [6], using queuing networks, we did a performance study of different cloud and fog configurations for sensing elderly vital signs. In another work [7], using Petri nets, we also focused on the server capacity, but now with sensors located inside and outside a hospital. In neither case were the sensors the focus of the study. Therefore, sensors can have a huge configuration variation that makes their operation unpredictable. Furthermore, sensors can fail, and only an individual analysis per sensor can provide a more realistic view of their performance.

Given this high unpredictability, adopting an assessment/method that guides minimum settings per sensor is necessary. Stochastic Petri Nets (SPNs) can be used to model and predict the performance of LoRaWAN sensors probabilistically. SPNs are known for their high degree of representativeness, being more intuitive than conventional options, such as Markov chains, to represent concurrency, parallelism, and synchronization in systems [8–10]. Some works use analytical models related to the problem addressed here but focus on the server layer. This paper presents an SPN model to model a LoRaWAN sensor network, including the network server layer. The most specific contributions of this work are:

- **Evaluation of sensors at the individual level**. To obtain detailed performance information, we modeled the sensors to identify each sensor's metrics, detailing their sensitivity, transmission capacity, and possible peculiarities in their responses to stimuli.
- **Fault tolerance analysis with retransmission mechanism**. The aim is to understand how the architecture responds to data-sending failures, examining the effectiveness and efficiency of the retransmission mechanism. This analysis aims to identify behavior patterns in the face of failures, contributing to improving the architecture and promoting greater robustness and reliability in environments prone to communication failures.
- **Focus on flow metrics, as it represents the impact of individual sensor characteristics**. The aim is to explore the impact of each sensor on the system flow. This approach aims to deepen the understanding of the direct impact of sensor particularities on the efficiency of the architecture, contributing to a more comprehensive and informed analysis.

This paper is organized as follows. Section 2 presents basic concepts about SPN. Section 3 presents the work related to the proposal of this article. Section 4 presents the architecture that served as the basis for building the model. Section 5 details the proposed base model, as well as an explanation of its components and activity flow. Section 6 presents the sensitivity analysis performed on the SPN model. Section 7 addresses the scenarios studied and analyzes the results obtained. Finally, Sect. 8 presents the conclusions and outlines future work.

## 2 Stochastic petri nets

Before describing the proposed model, we will start with a quick overview of stochastic Petri net models. SPNs have transitions that can be immediate and timed [11]. Timed transitions have an exponentially distributed delay, while immediate transitions have an associated delay time equal to zero, that is, they are performed instantly as soon as they are enabled. However, when more than one immediate transition is enabled simultaneously, a tiebreaker is required to determine which transition will be triggered first. These rankings are determined by the weights assigned to the transitions. [12]. SPNs enable probabilistic systems modeling and analysis. The memoryless property of the exponential distribution in shot delay implies that SPNs are isomorphic to continuous time Markov chains (CTMCs), thus providing measures of performance and dependability [13]. We are using the Generalized Stochastic Petri Net (GSPN) as an extension of the Stochastic Petri Net [14]. GSPNs are suitable for representing dynamic systems with discrete events because they have different types of timed transitions, but for simplification, we will call them SPNs in the text.

Figure 1 presents a simple example of an SPN and its main components. Arcs (directed edges) connect places to transitions and vice versa. Tokens can reside in places that denote the state (i.e., tagging) of an SPN. Tokens typically represent new requests entering the system. In this paper, each token that arrives in the system means a new package to be delivered to the destination. The behavior of an SPN is defined in terms of a token flow. The performance of an action or event (firing a transition) in the system is linked to preconditions. Tokens are created and destroyed according to transition fire. Each transition has an associated time that follows a certain probability distribution. In the example in Fig. 1 there is a generation of new tokens in the T1 transition that follows a certain arrival rate. The tokens are then queued at place F, which does not have a maximum queuing limit. To actually start processing in the system, the token must undergo a period linked to the T2 transition. This entry will only occur if there is sufficient capacity in place C. C represents the maximum parallel performance capacity of the system. When the token reaches place P, it starts counting the transition time T3 and ends one cycle.

## 3 Related work

This section presents studies related to the theme of this work, as identified in the literature. The search covered papers that explore LoRaWAN architectures, focusing on aspects such as intergenerational time, failure rate, retransmission rate, and distance between sensors and the gateway and between the gateway and the server. The study by [15] proposes investigating the LoRaWAN radio channel in the 868 MHz band through extensive measurement campaigns in indoor and outdoor environments, covering both urban and rural locations. Using the empirical results of these measurements, they evaluated performance and devised path loss models for LoRaWAN communications, comparing them with widely adopted empirical models. The work of [16] proposes a secure and fault-tolerant architecture to address security and availability issues in LoRaWAN technology. Additionally, they conducted performance analysis of the blockchain network in a cloud environment, considering various workloads, and performed fault tolerance experiments to understand the impact of failures on the network.

The work of [17] proposes evaluating the performance of LoRaWAN technology using synchronized Aloha. To do this, they conduct extensive simulations, analyzing metrics such as packet error rate, throughput, delay, and energy consumption in different scenarios. The study by [18] proposes an analytical model to calculate the delay and energy consumed in the reliable delivery of Uplink data in Class A LoRaWAN networks. Practical application of the model includes evaluating the impact of the number of end devices and the maximum number of data frame retransmissions on confirmed delivery of Uplink data in LoRaWAN networks. The work by [19] seeks to evaluate the performance of the LoRaWAN network in urban environments, using several propagation models, and compare the simulation results with real measurements. The study by [20] proposes an analytical model for multi-route LoRaWAN networks and investigates energy consumption, throughput, delay, and packet delivery rate in single-hop relay and multi-route networks.

The work of [21] investigates the possibility of having confirmed traffic in LoRaWAN networks under unreported channel interference. The results were obtained through simulations using an open-source network simulator, providing information for LoRaWAN operators in optimizing their networks against interference before real deployments. The study by [22] proposes a comprehensive analysis of the known causes of packet loss in an uplink-only LoRaWAN network. Furthermore, they investigate duty cycle limitations, packet collisions, insufficient coverage, and saturation of receiver demodulation paths. The work of [23] proposes introducing a retransmission mechanism in the LoRaWAN application layer with Lorawan to guarantee the completeness of sensor data. The extension aims to improve communication robustness between end nodes and the network cloud, ensuring continuous and reliable data transmission.

Table 1 compares important points of this study with other studies in the area. The first comparative criterion is the evaluation method. The studies researched showed a slight tendency to use simulations to evaluate their proposals. However, our study uses SPN modeling to probabilistically predict some flow situations that can occur and provide adaptability in component choices for a real system. SPN modeling is well-used when the proposed system requires a high cost for real tests. The metrics evaluated in this context can be diverse, but the flow metric becomes crucial for a system of this type. The metrics studied help understand the system's behavior and parameterize the sending of packages. Unlike all other works, this study seeks to present an innovative stochastic model for evaluating the performance of LoRaWAN architectures, emphasizing packet throughput analysis and fault tolerance.

The proposal aims to capture behavior patterns, allowing the measurement of the impact of modifying parameters on the efficiency of the architecture. The research covers the generation interval, the failure rate, the retransmission rate, the distance between sensors and the gateway, and the distance between the gateway and the

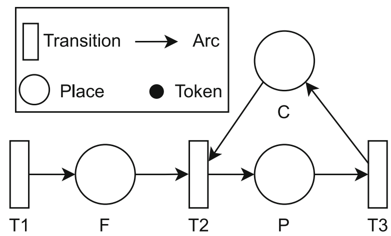



**Fig. 1** Simple example of an SPN

server. This approach stands out from other studies that incorporate all these concepts. The last factor compared is whether the study considers sensitivity analysis. No other work sought to use this mechanism to verify which factor impacts the system most in the research carried out.

## 4 Architecture

The LoRa specification includes two layers: the physical layer (LoRa RF) and the link layer, called LoRaWAN. Low energy consumption and communication over long distances are characteristics achieved by LoRa thanks to the Chirp Spread Spectrum (CSS) modulation scheme, used in the physical layer, and the use of LoRaWAN, based on pure ALOHA [25]. However, the increase in collisions due to the longer time in the air (Time on Air - ToA) of packet transmissions has become a disadvantage of the technology [26]. This behavior is related to the inadequate selection of transmission powers and scattering factors by end devices [27]. While higher transmission powers decrease

**Table 1** Related Work

| Works | Metrics | Method of assessment | Parameters configurable of simulation | Analysis of sensitivity |
|---|---|---|---|---|
| [15] | Signal strength | PL model | Time between generations | No |
| [16] | Latency and throughput | Experiment | Failure rate | No |
| [17] | Error rate, throughput | Simulation | Time between generations | No |
| [18] | Date delay and power consumption | Simulation | Rate of retransmission | No |
| [19] | Received signal strength | Model of propagation | Time between | No |
| [20] | Flow, delay and delivery fee packages | Simulation | Rate of | No |
| [21] | Performance | Simulation | Rate of retransmission | No |
| [22] | Packet loss and the weight of each condition interruption | Simulation | Time to streaming among nodes | No |
| [23] | Signal strength received, the relationship signal-to-noise ratio, the package delivery and the integrity rate of date | Experiment | Rate of retransmission | No |
| [24] | Response time and the overall network delay | Experiment | Time between generations | No |
| This work | Throughput | SPN Model | Time between generations, failure rate, Rate of retransmission and time of streaming among nodes | Yes |

opportunities for spatial reuse, higher scattering factors make the air time of transmitted signals longer [28]. Therefore, to minimize the probability of collisions and improve access to the medium, the appropriate configuration of spreading factors and transmission powers must be sought [29]. After making such configurations on the sensors, they must be positioned strategically to obtain the desired performance. A very important issue is the distance between each sensor and its gateway, as well as the distance from the gateway to the network server.

This section outlines the architecture adopted as the basis for creating the SPN model in question. Figure 2 illustrates the structure modeled in this paper. The architecture represents the individual communication of each sensor that sends its data through the gateway until it reaches the network server. The network is based on LoRaWAN technology, a network architecture in which centralized access points (gateways) manage the communication of LoRa devices with the Internet. One of the contributions of this paper is a fault tolerance analysis with a retransmission mechanism. When data is sent through the sensor but fails, it can be retransmitted to the destination or discarded, optimizing resource use and avoiding waste.

The sensors used in this system can be of different types, each designed to measure a specific physical parameter of the environment. Parameters may include temperature, relative humidity, wind speed, and gas concentration. LoRa nodes, which play a crucial role in collecting data through sensors, are responsible for gathering environmental information. The gateway collects data from the sensors, while the sensors are distributed throughout the scenario. Implementing the topology requires a design in which end devices utilize class A of the LoRaWAN specification, allowing devices to transmit data regularly. This choice was made because class A devices consume less energy. The solution adopted consists of the data being stored on a network server, which can provide the necessary tools to collect, process, and store the data from the sensors in a database later to ensure efficient data collection and storage.

## 5 Model

This section presents the SPN model[5] developed considering the peculiarities of the proposed architecture. Figure 3 shows the base model. The model is divided into three parts, namely: (1) Sensor, (2) Gateway, and (3) Network Server. Table 2 illustrates the main components of the model. This information is important for the reader to have an initial understanding of the model. Further on in

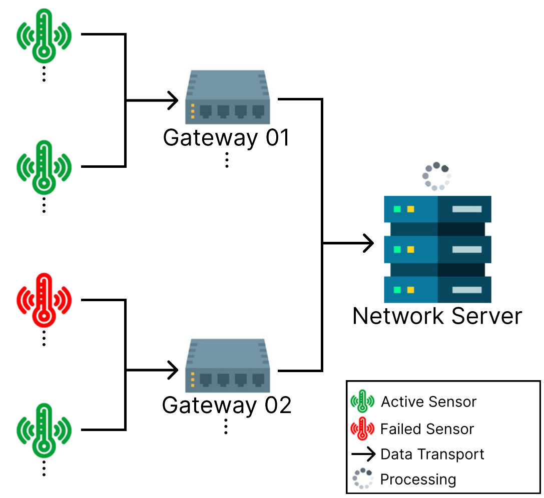



**Fig. 2** Proposed LoRaWAN architecture with fault tolerance

Table 5 it is possible to find vital data for the functioning of the model, containing values of places and transitions.

The first part of the model is made up of sensors. Each sensor contains three parts: the input, the transmission, which is part of the sensor itself, and the relay. The input is composed of two transitions, $T_{adx}$ and $T_{dgx}$, and a place $P_{adx}$, where x indicates the sensor number. $T_{adx}$ is a deterministic time distribution transition and represents the time between the arrival of requests at each sensor. When $T_{adx}$ is fired, a *token* is deposited in place $P_{adx}$, which represents the arrival of requests at the sensor. $T_{dgx}$ represents the average time taken to transmit a request, and when fired, it deposits a *token* in place of $P_{sfx}$. $P_{sfx}$ is an auxiliary place to indicate whether the transmission succeeded or failed. If the transmission occurs successfully, the transition $T_{ws}$ is activated, and the token is deposited in place $P_{gx}$. If the transmission fails, the transition $T_{wf}$ is fired, and the *token* is deposited at the place $P_{drx}$, where the retransmission will take place.

The place $P_{drx}$ represents the state where an attempt to reconnect and retransmit with the gateway is made. The $T_{wd}$ transition is fired if the connection fails, and the *token* is discarded. If the connection is successfully reestablished, the transition $T_{wp}$ is fired, and the guard condition `N_RET` is triggered, making the multiplicity of replicas, and the token is deposited in place $P_{rx}$. The place $P_{rx}$ represents the state where the sensor relays the request. The transition $T_{drgx}$ represents the request retransmission time, and when it is activated, the *token* is deposited in place $P_{gx}$. The second part of the model is made up of gateways. The gateways receive requests from $n$ sensors. In the figure shown, each gateway receives requests from two sensors. The first place in the component is $P_{gx}$, which represents the input queue of gateway x. In this model, the gateway is assumed to have a large queuing capacity, so a

[5] https://github.com/IsraelCardoso/LoRaWANSPNModel.

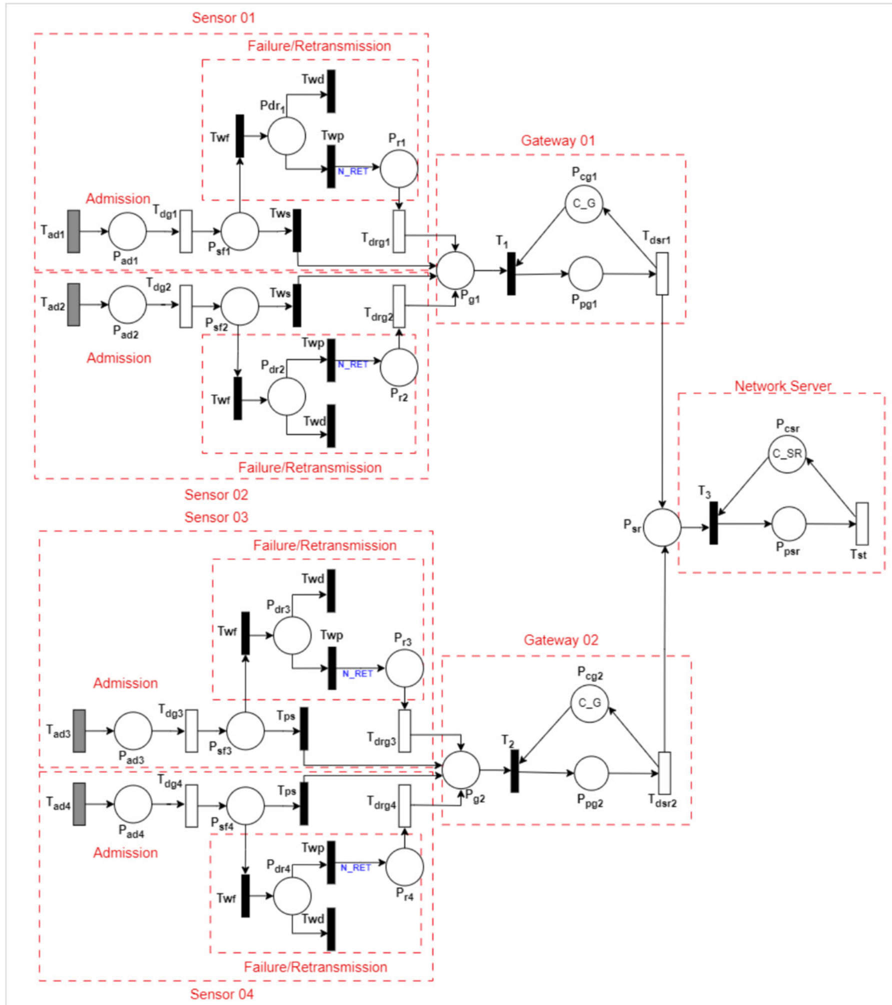



**Fig. 3** SPN model developed from the LoRaWAN architecture

queue of limited size was not considered for storing requests.

When a request arrives at the gateway and there is processing capacity available in place $P_{cgx}$, the transition $T_x$ is fired and a *token* is deposited in place $P_{pgx}$. The place $P_{pgx}$ represents the state where the request is forwarded to the server. The transition $T_{dsrx}$ represents the time it takes the gateway to transmit a request to the gateway, and when fired, it deposits the *token* in place $P_{sr}$. The network server component forms the third part of the model. The place $P_{sr}$ represents the server's input queue. Like the gateway, the server's request queuing capacity was not limited. When there is processing capacity on the server, represented by place $P_{csr}$, the request is forwarded for processing by activating the $T_3$ transition. The place $P_{psr}$ represents the state where the server processes the request. Finally, the transition $T_{st}$ represents the time the server needs to process a request.

The metric used in this study was *Throughput* (TP), a crucial quantitative measure for assessing a system's ability to process requests and transfer data. Within the scope of the model being evaluated, TP is based on Little's Law [30], which is calculated by considering the expected number of requests being processed divided by the processing time of each request. The equation (1) presents the syntax of the metric used in Mercury [31]. It is measured by indicating the number of requests successfully processed by the system in a given period. TP is calculated by the expectation of having *tokens* in the queue representing the processing location of the network server, divided by the time it takes for the request to be processed. Where $E\#Ppsr$ represents the expected number of *tokens* in a given period. In this equation, "E" denotes "*Expected*", indicating an estimate or expected average. The value of *Tst* represents the time it takes for the system to process the *token*. Throughput is relevant in this scenario because it indicates the architecture's ability to expand to meet the growing traffic demand. Throughput becomes relevant in this context because it allows us to identify the system's performance in relation to changes in the various factors to

**Table 2** Description of the main components of the model

| Type | Component | Definition |
|---|---|---|
| Places | Pad1, Pad2, Pad3, Pad4 | New packages in the system |
| | Psf1, Psf2, Psf3, Psf4 | Queuing packets for the gateway |
| | Pdr1, Pdr2, Pdr3, Pdr4 | Queuing packets for discard or retransmission |
| | Pr1, Pr2, Pr3, Pr4 | Queuing packets for retransmission |
| | Pg1, Pg2 | Gateway processing capacity |
| | Pcg1, Pcg2 | Queue at packet collection gateway |
| | Ppg1, Ppg2 | Gateway Capacity Queue |
| | Psr | Gateway queue |
| | Pcrs | Network Server Capacity Queue |
| | Ppsr | Network Server Queue |
| Deterministic transitions | Tad1, Tad2, Tad3, Tad4 | Package arrival time |
| Immediate transitions | Twf | Probability of failed transmission |
| | Tws | Probability of successful transmission |
| | Twd | Probability of transmission with discard |
| | Twp | Retransmission probability |
| | T1, T2, T3 | Time between sensor packet arrivals |
| Exponential transitions | Tdg1, Tdg2, Tdg3, Tdg4 | Transmission time from node to gateway |
| | Tdrg1, Tdrg2, Tdrg3, Tdrg4 | Retransmission time from node to gateway |
| | Tdrs1, Tdrs2 | Transmission time from gateway to network server |
| | Tst | Packet transition time on the network server |
| Marking places | AD | Arrival of packages |
| | C_G | Gateway Capacity |
| | C_SR | Network Server Capacity |

which the system is subject, such as variation in errors, increased delay in message delivery, increased number of messages in the system, among others.

$$\mathbf{TP} = \frac{E\{\#Ppsr\}}{Tst} \tag{1}$$

## 6 Design of experiments

In this section, we will use the SPN model to analyze the elements that can affect the performance of the system under study. Stochastic Petri Nets are a powerful tool for modeling and analyzing complex systems, enabling a comprehensive understanding of their behavior and performance. Our main focus will be on the throughput metric, which also represents the impact of the individual characteristics of the sensors. By understanding how different components interact and influence TP, we can identify opportunities for improvement and optimization [32].

To conduct this analysis, we will describe in detail the structure of the system under study, considering the different processes, resources, and events that make up its functioning. We will also analyze the interactions between such elements and how they affect the overall TP of the system. Furthermore, we discuss the results obtained from applying the SPN model. Based on these analyses, we will gain valuable information into how certain modifications to system components can impact performance, allowing us to propose specific solutions and improvements.

Next, we will address the Design of Experiments (DoE) method, a systematic and statistical approach to investigating the effect of key variables in a system. DoE is extremely useful for exploring different scenarios and obtaining accurate information about how variables influence system performance. The combination of the SPN model and the DoE will help us understand and base our optimization recommendations on a robust and well-founded basis.

### 6.1 Design of experiments overview

The present study employs advanced DoE techniques to perform a comprehensive sensitivity analysis. DoE, or the design of experiments, is a structured process of planning, designing, and analyzing experiments that aim to efficiently and effectively obtain valid and objective conclusions. This approach involves performing a series of tests in which a research engineer modifies the input variables or factors to identify the causes of the changes observed in the output response [33].

It is essential to highlight that DoE techniques provide a systematic and structured approach to sensitivity analysis. These techniques allow a comprehensive assessment of the factors that influence the system under study, allowing the identification of the most significant factors and their impact on the output response. Based on the graphic categories recommended by specialized literature, the aim is to provide an intuitive visual representation of the results, facilitating data interpretation and the decision-making process [34].

The DoE analysis performed in this study is based on two main graph categories: factor interaction graphs and effects graphs. Interaction graphs are a valuable tool for visualizing how factors interact with each other [35]. They use lines to represent interactions between factors, indicating whether one factor can affect the outcome of another. When the lines are parallel, it is suggested that there is no interaction between the factors. However, if the

**Table 3** Design of Experiments

| Factor name | Low setting | High setting |
|---|---|---|
| C_G | 50.0 (qtd) | 150.0 (qtd) |
| C_SR | 10.0 (qtd) | 100.0 (qtd) |
| Trasmission time | 1000.0 (ms) | 3000.0 (ms) |
| Twf | 0.1 (%) | 0.9 (%) |
| Twp | 0.1 (%) | 0.9 (%) |

**Table 4** Combination of factors considering the throughput metric

| C_G(qdt) | C_SR(qdt) | Transmission time(ms) | Twf(%) | Twp(%) | TP(pct/ms) |
|---|---|---|---|---|---|
| 50.00 | 10.00 | 1000.00 | 0.10 | 0.10 | 0.002175519 |
| 50.00 | 10.00 | 1000.00 | 0.10 | 0.90 | 0.003225104 |
| 50.00 | 10.00 | 1000.00 | 0.90 | 0.10 | 0.000694403 |
| 50.00 | 10.00 | 1000.00 | 0.90 | 0.90 | 0.003270508 |
| 50.00 | 10.00 | 3000.00 | 0.10 | 0.10 | 0.001070262 |
| 50.00 | 10.00 | 3000.00 | 0.10 | 0.90 | 0.001836415 |
| 50.00 | 10.00 | 3000.00 | 0.90 | 0.10 | 0.000773032 |
| 50.00 | 10.00 | 3000.00 | 0.90 | 0.90 | 0.001109548 |
| 50.00 | 100.00 | 1000.00 | 0.10 | 0.10 | 0.002118254 |
| 50.00 | 100.00 | 1000.00 | 0.10 | 0.90 | 0.002986661 |
| 50.00 | 100.00 | 1000.00 | 0.90 | 0.10 | 0.001462720 |
| 50.00 | 100.00 | 1000.00 | 0.90 | 0.90 | 0.003804096 |
| 50.00 | 100.00 | 3000.00 | 0.10 | 0.10 | 0.001196712 |
| 50.00 | 100.00 | 3000.00 | 0.10 | 0.90 | 0.001361617 |
| 50.00 | 100.00 | 3000.00 | 0.90 | 0.10 | 0.000603426 |
| 50.00 | 100.00 | 3000.00 | 0.90 | 0.90 | 0.001192422 |
| 150.00 | 10.00 | 1000.00 | 0.10 | 0.10 | 0.002481466 |
| 150.00 | 10.00 | 1000.00 | 0.10 | 0.90 | 0.002993262 |
| 150.00 | 10.00 | 1000.00 | 0.90 | 0.10 | 0.001253853 |
| 150.00 | 10.00 | 1000.00 | 0.90 | 0.90 | 0.003546888 |
| 150.00 | 10.00 | 3000.00 | 0.10 | 0.10 | 0.001092394 |
| 150.00 | 10.00 | 3000.00 | 0.10 | 0.90 | 0.001390318 |
| 150.00 | 10.00 | 3000.00 | 0.90 | 0.10 | 0.000551892 |
| 150.00 | 10.00 | 3000.00 | 0.90 | 0.90 | 0.000960975 |
| 150.00 | 100.00 | 1000.00 | 0.10 | 0.10 | 0.002521636 |
| 150.00 | 100.00 | 1000.00 | 0.10 | 0.90 | 0.003334495 |
| 150.00 | 100.00 | 1000.00 | 0.90 | 0.10 | 0.001274768 |
| 150.00 | 100.00 | 1000.00 | 0.90 | 0.90 | 0.003675449 |
| 150.00 | 100.00 | 3000.00 | 0.10 | 0.10 | 0.000915030 |
| 150.00 | 100.00 | 3000.00 | 0.10 | 0.90 | 0.001569367 |
| 150.00 | 100.00 | 3000.00 | 0.90 | 0.10 | 0.000471951 |
| 150.00 | 100.00 | 3000.00 | 0.90 | 0.90 | 0.001248960 |

lines are not parallel, the factors interact, jointly influencing the results.

On the other hand, the effects graph is a technique that uses bars ordered in descending order to represent the impact of factors on the analyzed measure. The larger the bar, the greater the influence of the corresponding factor. This approach is particularly useful for identifying and prioritizing the most important factors, allowing for efficient resource and effort allocation. The effects graph is often used to maximize impact, focusing on factors with the greatest potential to influence the analyzed measure positively.

The experiment was conducted using the proposed architecture and its components. The sensitivity analysis examined these components; however, only the factors with significant interactions are discussed. The interaction between factors was evaluated based on their impact on the TP metric, selected for its direct relevance to end-user perception. The factors analyzed in this study include five system components: (i) *C_G*, (ii) *C_SR*, (iii) *Transmission Time*, (iv) *Twf*, and (v) *Twp*. These factors were evaluated at two configuration levels: low and high. The factors have two levels, which are low and high settings. A Table 6.1 presents the values generated from a default value, considering a 50% increase (maximum value) and a 50% reduction (minimum value). Table 6.1 illustrates the most relevant combinations of these values presented in Table 6.1 and displays the results of the 32 combinations related to the combinations used to create the DoE. The combination of these factors allows the application of the DOE technique, which makes it possible to identify which components of the architecture are most relevant and how their variations can impact the results (Tables 3, 4).

## 6.2 Results analysis

This subsection presents the results of the sensitivity analysis with DoE. Different scenarios considering varied parameter combinations were analyzed in order to evaluate the impact on the metric. It is important to highlight that even after increasing resources, such as the*C_G* gateway capacity, some scenarios may not demonstrate significant gains in throughput. This behavior may be justified by factors such as the specific configuration of parameter combinations, which may not fully take advantage of the additional features. The main objective of DoE is exactly to identify scenarios in which the impact of adjustments is more evident, allowing us to understand the limits and behaviors of the system. Fig. 4 presents the effect graph of the factors. The influence of each factor individually and the effects of the interactions between the factors make up the graph. The factor *Transmission time* had the greatest impact on TP, followed by the factor *Twp*. The *Transmission Time* factor had an impact greater than 0.0014, and *Twp* had an impact of ≈ 0.0010. The *Twf* The other interactions and factors had a smaller impact on the TP metric than those mentioned previously. The results are relevant for developing strategies to improve PT. The data presented allows the system architect to identify and prioritize the most critical factors.

Fig. 5a shows the interaction between *C_G* and *C_SR* with respect to TP. The combination *C_SR* = 10 and *C_G* = 50 achieves the highest TP, indicating that lower capacities were sufficient to meet system demand. The *C_SR* factor has the greatest impact, as evidenced by the slope of the blue line (*C_SR* = 10), while *C_G* shows minimal variation. At higher levels of *C_SR*, the TP remains stable, even with the increase of *C_G*, which suggests saturation of the system. This stability can be advantageous for scenarios that prioritize reliability, while lower capacities favor greater sensitivity and fine-tuning in performance. Fig. 5b illustrates the interaction between *Transmission Time* and *Twf* in the TP metric. A significant interaction between the factors is observed, as evidenced by the inverse behavior of the lines. It is worth noting that an increase in *Transmission Time* does not necessarily imply a degradation in TP performance. For *Twf* = 0.9, the TP is higher when *Transmission Time* is at level 1000, whereas for *Twf* = 0.1, the TP improves with *Transmission Time* = 3000. The combination that achieved the highest TP was *Transmission Time* = 3000 and *Twf* = 0.1, with TP reaching approximately 0.00192. These results suggest that the impact of *Transmission Time* on throughput strongly depends on the value of *Twf*, highlighting the importance of considering both factors jointly to optimize system performance.

Figure 5c illustrates the interaction between *Transmission Time* and *Twp* in relation to TP. A significant

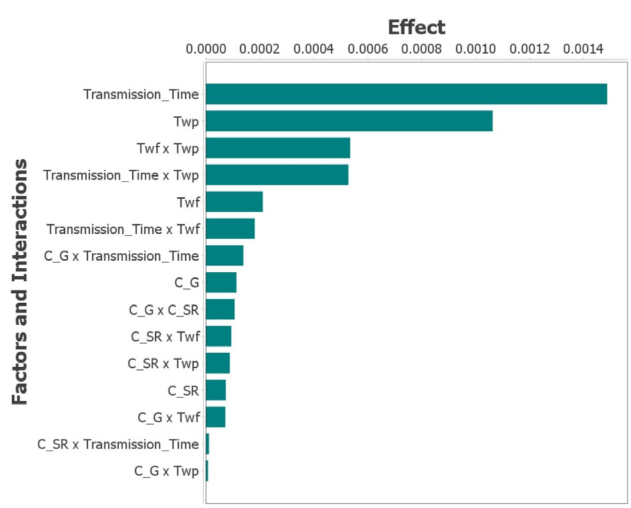
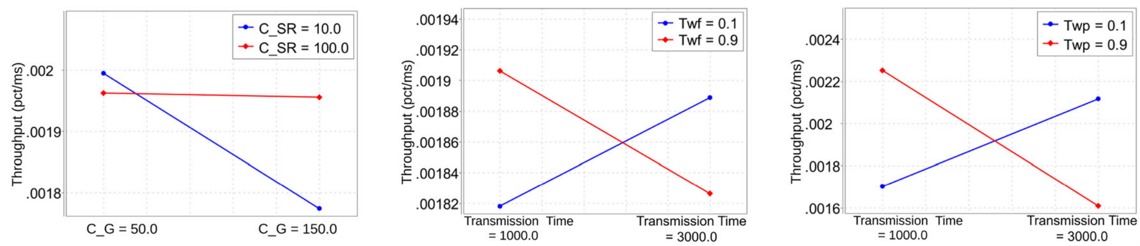



**Fig. 4** Visualization of the impact of various model factors on the throughput metric




(a) Gateway capacity vs. Network Server capacity.

(b) Transmission Time vs. Failuire Time.

(c) Transmission Time vs. Retransmission Time.

**Fig. 5** Interaction between factors regarding their impact on the throughput metric

interaction between the factors is observed, as evidenced by the crossing of the lines. When *Twp* = 0.9, the TP is higher for *Transmission Time* = 1000, reaching approximately 0.0022. On the other hand, for *Twp* = 0.1, the TP increases with higher levels of *Transmission Time*, being highest at *Transmission Time* = 3000. These results show that the behavior of *Transmission Time* in relation to TP is directly influenced by the level of *Twp*. While higher levels of *Twp* favor better performance at shorter transmission times, lower levels of *Twp* benefit from longer transmission times.

However, it is important to note that both factors have similar impacts on the TP metric, which is evidenced by the similar slopes of the lines in the graph. Lines with similar slopes indicate that variations in both factors influence the TP proportionally, indicating that adjustments in either of the two factors can result in equivalent impacts on network performance. This similarity is evidence that the factors have a similar relationship with the system response, which reinforces the importance of analyzing them together when developing optimization strategies.

## 7 Impact analysis of different metrics

This section presents the results generated with the proposed model. Due to the complexity of the model, the analyses were conducted using stationary simulations, with an approximate margin of error of 2%. The Mercury tool[6] was used to create the model and generate the results [36]. Table 5 details the values that feed the model for all case studies. These values were derived by carrying out practical experiments. The transitions $T_{adx}$ and $T_{dgx}$ are the only ones of type *single server*; the rest are of type *infinite server*. The subsequent subsections show five case studies as simulation results. All results focus on the system's data throughput. In the first four results, the number of nodes in the network varied (from 1 to 4), considering the equidistant Transmission Time between the nodes (N), the gateway (G), and the server (S) (maintaining the same Transmission Time, N-G =G-S). The limitation of four nodes was established to facilitate the analysis of the proposed scenario. It is important to highlight that the model was designed to be scalable, allowing adaptations as needed to evaluate scenarios with a larger number of nodes.

timetimetime

### 7.1 Case study 1 - time between generations

This section presents the average time between generations of the system, which refers to the time interval between the successive sending of packets by the nodes of a network. This concept is equivalent to the packet arrival time, which represents the period between the generation of one packet and the next. The analysis highlights the relevance of the variation in the rate between generations to understand the results achieved. Manipulating the number of nodes in the network (from 1 to 4) revealed the interaction between the generation rate and the time between generations. Figure 6 visualizes the impact of this variation in the time between generations, playing a crucial role in directly influencing the flow of the system.

Figure 6 shows the impact of the variation in time between generations. As $T_{adx}$ is increased, a decrease in throughput is observed, indicating that an increase in the time between boosts effectively reduces the amount of data delivered to the destination. This update translates into a progressive decrease in the performance of the communication lines as the interval between the source and the destination becomes longer. The influence of the number of nodes in the network is also evident, with an increase in the number of nodes contributing to an increase in throughput. However, after 1,000 milliseconds, there is a reduction in TP. This is caused by the increase in the waiting time between specifics, which leads to a saturation of the network's processing capacity. When the packet arrival time $T\_adx$ exceeds certain limits, such as 1,000 milliseconds, a reduction in TP is observed. This indicates that longer arrival intervals do not overload the system but instead

[6] https://www.modcs.org/

**Table 5** Parameters used in the model

| Component | Value | Weight(%) | Description |
|---|---|---|---|
| Tadx | 1.500 (ms) | - | Package arrival time |
| Tdgx | 1.000 (ms) | - | Transmission time from node to gateway |
| Tdrgx | 1.000 (ms) | - | Retransmission time from node to gateway |
| Tdsrx | 1.000 (ms) | - | Transmission time from gateway to network server |
| Tx | - | - | Time between sensor packet arrivals |
| Pws | - | 90 | Probability of successful transmission |
| Pwf | - | 10 | Probability of failed transmission |
| Pwd | - | 90 | Probability of transmission with discard |
| Pwp | - | 10 | Probability of retransmission |
| N-RET | 5 (qtd) | - | Number of packets retransmitted |
| C_G | 50 (qtd) | - | Gateway Capacity |
| C_SR | 10 (qtd) | - | Network Server Capacity |
| Tst | 8 (ms) | - | Network Server service time |

reduce the frequency of processed packets, which directly impacts overall performance. As shown in the graph, throughput consistently decreases as the time between generations increases, regardless of the number of nodes. This change of scenario suggests a tipping point. Where the network reaches its maximum capacity and, from that point on, the increase in the time between specific connections begins to impact TP levels. The increase in throughput with the expansion of the number of nodes in the network is justified by the increase in communications from multiple sources.

### 7.2 Case study 2 - average transmission time from the node to the gateway and from the gateway to the network server

Figure 7 shows the impact of varying the Transmission Time between network components and the system flow. The results are relative to varied configurations of the parameters $T_{dgx}$, $T_{drgx}$, and $T_{dsrx}$ in order to consider the impact of Transmission Time on transmission time. The parameters $T_{dgx}$, $T_{drgx}$, and $T_{dsrx}$ were manipulated at specific intervals, varying between 1,000 ms, 2,000 ms, 3,000 ms, 4,000 ms, and 5,000 ms. A tendency towards a decrease in throughput was observed as the node transmission time increased. A significant reduction stands out, especially when using 4 nodes, where the drop is significant. From 1,200 ms onwards, this decrease becomes more pronounced, highlighting a critical point at which the increase in $T_{dgx}$, $T_{drgx}$ and $T_{dsrx}$ has a considerable impact on transmission efficiency. More nodes in the network increase throughput, indicating a positive relationship between the number of nodes and system efficiency.

### 7.3 Case study 3 - failure rate

Figure 8 shows the impact of $T_{wf}$ on the system. In this study, the weight assigned to failures was adjusted to analyze the effect of failure probabilities on system performance in different node configurations. In single-node systems, TP decreases progressively as the failure weight increases, but the magnitude of this reduction is relatively smaller compared to systems with more nodes. This occurs because the performance depends solely on the single node, which continues transmitting data, even if in a limited manner, due to the lack of redundancy. In multi-node systems, as shown for configurations with 2, 3, and 4 nodes, the impact of failures is more pronounced. However, the use of multiple sensors in the same area provides greater redundancy, partially offsetting the increase in failure probabilities. For instance, even with a failure probability of 90%, systems with 4 nodes still maintain a throughput significantly higher than that of systems with only 1 node. The study also revealed a general trend of decreasing throughput as $T_{wf}$ increases, suggesting an inversely proportional relationship. In multi-node systems, this trend is less linear due to interactions between nodes and redundancy, which introduce variations in behavior. This pattern demonstrates that increasing the failure weight is associated with a decrease in data transmission efficiency, highlighting the importance of considering the number of nodes when designing systems for environments with high failure rates.

### 7.4 Case study 4 - retransmission rate

Figure 9 shows the impact of varying the retransmission rate on the system throughput. The result refers to $T_{wp}$, with $T_{wf}$ adjusted to 0.5 to induce higher failures, increasing retransmissions. The retransmission weight was varied by

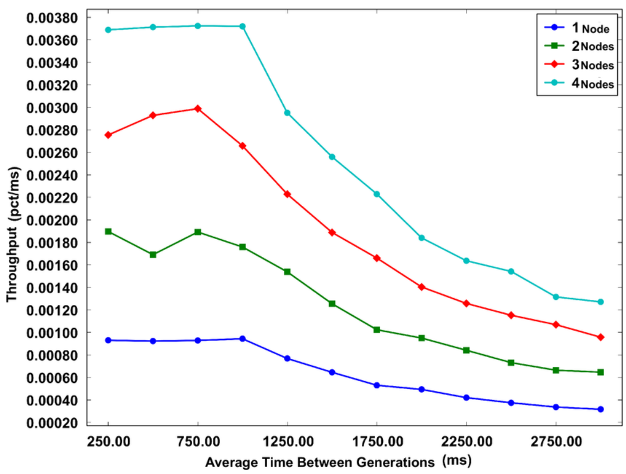



**Fig. 6** Time between generations

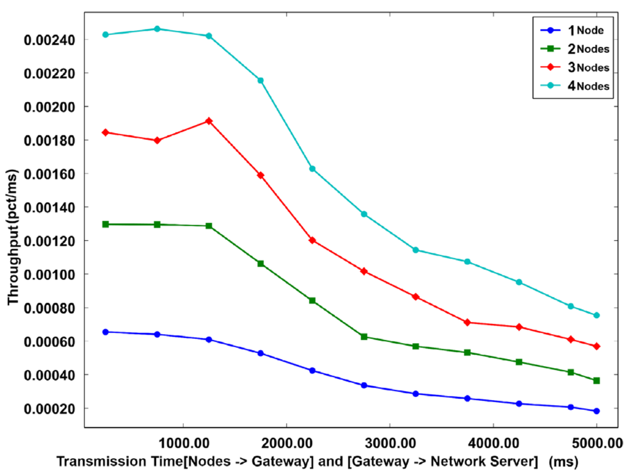



**Fig. 7** Transmission time from the node to the gateway and from the gateway to the network server

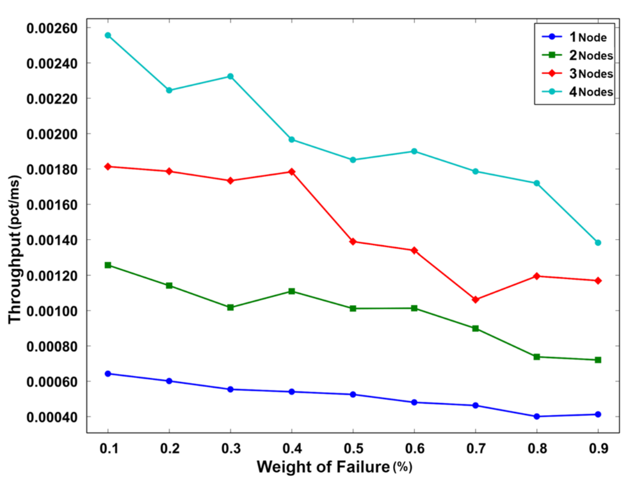



**Fig. 8** Variation weight of failure

0.1, 0.5, and 0.9, with the change in the number of retransmissions ranging from 1 to 10. Evidence exists between the number of retransmissions and the probability of additional details occurring. As the retransmission weight increases, the system has a greater throughput. This is expected, as more packages are traveling through the system, which increases the chance of delivering information but can also overload the system. The TP is small when the retransmission weight is low, specifically 0.1. However, when the value increases to intermediate values, such as 0.5, and higher values, such as 0.9, a tendency to bring the transmission lines closer together is observed. This behavior suggests that as the number of retransmissions increases, the differences in throughput between different retransmission weights become smaller.

## 7.5 Case study 5 - failure time x retransmission time

This section presents the result of the Failure Time (FT) variation in relation to the Retransmission Time (RT) for the system's throughput. Figure 10 shows the result of varying FT in conjunction with RF for system throughput. For this analysis, an FT range was defined between 0.1 and 0.9. The RT was set for the analysis, ranging between 0.1 and 0.9. These values can be found in Table 5 in the Twf and Twp transitions.

The graph displays the increase in Throughput as FT and RT increases. When the maximum failure rate exists (in this FT analysis =0.9 and RT=0.9) Throughput is greater than 0.010 ms. When rates start to be reduced, throughput tends to decrease. This is due to the reduced number of system failures, impacting its performance. In the best-case scenario, with FT=0.1 and RT=0.9, throughput has a minimum peak of 0.002ms. Therefore, it was observed that

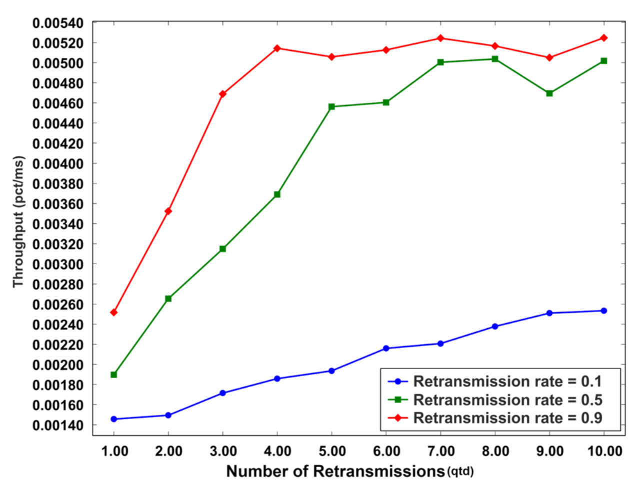



**Fig. 9** Impact of varying the retransmission rate on the system throughput

as the failure rate increases, more packets will be processed, and as a result of this increase, the system's throughput tends to be higher.

In this analysis, the graph displays useful information for the evaluator in an environment with constant variation, allowing the system to understand behavior in a given circumstance. For example, increasing the number of retransmissions will reduce the failure rate, increasing system throughput. In the tests carried out for this graph, the growth in throughput is not excessive, but it indicates that fault tolerance guarantees the system's performance to avoid wasting resources. The results corroborate previous DoE findings that FT and RT impact system capacity. If this capacity is too low, it may harm system performance by not passing enough packets. Conversely, increasing capacity at a time of low packet traffic will waste resources.

## 8 Conclusion

In this work, an SPN model was proposed to evaluate the performance of a fault-tolerant LoRaWAN architecture. The SPN model developed in this research calculates the key metric TP to evaluate the system performance and throughput by measuring the efficiency of packet forwarding, retransmissions, and failures in the transmission and execution of requests by each sensor. During the simulations, it was observed that when the time between packet generations ($T_{adx}$) exceeds 1,000 ms, the system becomes overloaded, leading to a reduction in throughput. On the other hand, increasing the number of nodes in the network resulted in an improvement of up to 30% in TP before reaching the performance limit. The paper presented five case studies that show how administrators can use the model to analyze different scenarios. These case studies highlight the performance gains obtained by changing each parameter. The results of this study provide guidance on optimal scenarios and configurations to avoid wasting resources in real applications. In future work, we plan to conduct tests for this type of architecture using other metrics such as latency and energy efficiency in order to obtain a more comprehensive evaluation of the system. Additionally, a validation experiment will be carried out to verify the accuracy of the model in real situations, addressing critical aspects of architecture optimization, such as scalability and retransmission management, ensuring that the system maintains high efficiency even in adverse scenarios, which will be discussed in detail later.

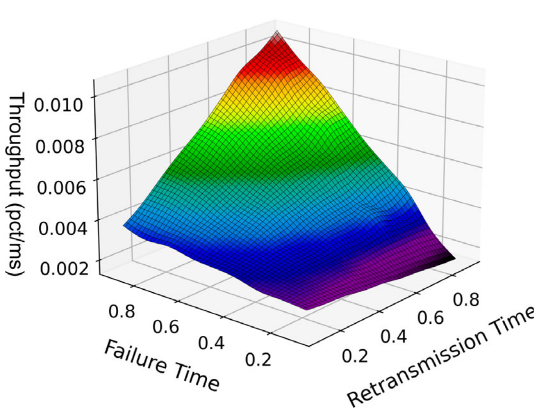



**Fig. 10** Simultaneous variation of failure time and retransmission time

**Author contributions** The authors have contributed equally.

**Funding** No funding was received for conducting this study.

**Data availability** Data sharing is not applicable.

## Declarations

**Conflict of interest** The authors have no relevant financial or non-financial interests to disclose.

**Ethical approval** Not applicable

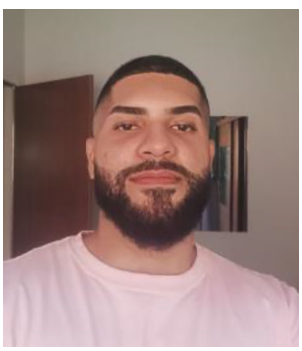
## References

1. Hessel, Frank, Almon, Lars, Hollick, Matthias: Lorawan security: an evolvable survey on vulnerabilities, attacks and their systematic mitigation. ACM Trans. Sensor Netw. **18**(4), 1–55 (2023)
2. Brito, Thadeu, Azevedo, Beatriz Flamia, Mendes, João., Zorawski, Matheus, Fernandes, Florbela P., Pereira, Ana I., Rufino, José, Lima, José, Costa, Paulo: Data acquisition filtering focused on optimizing transmission in a lorawan network applied to the wsn forest monitoring system. Sensors **23**(3), 1282 (2023)
3. Piechowiak, Maciej, Zwierzykowski, Piotr, Musznicki, Bartosz: Lorawan metering infrastructure planning in smart cities. Appl. Sci. **13**(14), 8431 (2023)
4. Loubany, Ali, Lahoud, Samer, Samhat, Abed Ellatif, Helou, Melhem El: Throughput improvement for lorawan networks considering iot applications priority. In 2023 6th Conference on Cloud and Internet of Things (CIoT), pages 206–210, (2023)
5. Bonilla, Vicky, Campoverde, Brandon, Yoo, Sang Guun: A systematic literature review of lorawan: sensors and applications. Sensors **23**(20), 8440 (2023)
6. Rodrigues, Laécio., Rodrigues, Joel JPC.., de Barrosantonio, Serra, Silva, Francisco Airton: A queueing-based model performance evaluation for internet of people supported by fog computing. Future Internet **14**(1), 23 (2022)
7. Rocha, Francisco, Nogueira, Bruno, Gonçalves, Glauber, Silva, Francisco Airton: Smart hospital patient monitoring system aided by edge-fog-cloud continuum: a performability evaluation focusing on distinct sensor sources. J. Netw. Syst. Manage. **32**(4), 99 (2024)
8. Silva, Luis Guilherme, Cardoso, Israel, Brito, Carlos, Barbosa, Vandirleya, Nogueira, Bruno, Choi, Eunmi, Nguyen, Tuan Anh, Min, Dugki, Lee, Jae Woo, Silva, Francisco Airton: Urban

advanced mobility dependability: a model-based quantification on vehicular ad hoc networks with virtual machine migration. Sensors **23**(23), 9485 (2023)
9. Fé, Iure: Tuan Anh Nguyen, André Soares, Seokho Son, Eunmi Choi, Dugki Min, Jae-Woo Lee, and Francisco Airton Silva. Model-driven dependability and power consumption quantification of kubernetes based cloud-fog continuum, IEEE Access (2023)
10. de Sousa, Rubenilson, Cristian, Leonardo, Feitosa, Leonel, Choi, Eunmi, Nguyen, Tuan Anh, Min, Dugki, Silva, Francisco Airton: Performability evaluation and sensitivity analysis of a video streaming on demand architecture. Appl. Sci. **13**(2), 998 (2023)
11. Silva Francisco Airton, Iure Fé, Gonçalves, Glauber: Stochastic models for performance and cost analysis of a hybrid cloud and fog architecture. J. Supercomput. (2020)
12. Carvalho, Daniel, Rodrigues, Laécio, Endo, Patricia Takako, Kosta, Sokol, Silva, Francisco Airton: Mobile edge computing performance evaluation using stochastic petri nets. In 2020 IEEE Symposium on Computers and Communications (ISCC), pages 1–6. IEEE, (2020)
13. Maciel, Paulo Romero Martins: Performance, reliability, and availability evaluation of computational systems, Volume 2: Reliability, availability modeling, measuring, and data analysis. CRC Press, (2023)
14. Sachan, Swati: Stochastic simulation: An introduction to generalized stochastic petri net (gspn). Medium, (2024)
15. El Chall, Rida, Lahoud, Samer, El Helou, Melhem: Lorawan network: radio propagation models and performance evaluation in various environments in lebanon. IEEE Intern. Things J. **6**(2), 2366–2378 (2019)
16. Ribeiro, Victor, Holanda, Raimir, Ramos, Alex, Rodrigues, Joel JPC.: A fault-tolerant and secure architecture for key management in lorawan based on permissioned blockchain. IEEE Access **10**, 58722–58735 (2022)
17. Ali, Zulfiqar, Henna, Shagufta, Akhunzada, Adnan, Raza, Mohsin, Kim, Sung Won: Performance evaluation of lorawan for green internet of things. IEEE Access **7**, 164102–164112 (2019)
18. Khan, Furqan Hameed, Jurdak, Raja, Portmann, Marius: A model for reliable uplink transmissions in lorawan. In 2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS), pages 147–156. IEEE, (2019)
19. Ingabire, Winfred, Larijani, Hadi, Gibson, Ryan M: Performance evaluation of propagation models for lorawan in an urban environment. In 2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE), pages 1–6. IEEE, (2020)
20. Rodrigues Cotrim, Jeferson, Kleinschmidt, João Henrique: An analytical model for multihop lorawan networks. (2021)
21. Martinez, Ivan, Nouvel, Fabienne, Lahoud, Samer, Tanguy, Philippe, El Helou, Melhem: On the performance evaluation of lorawan with re-transmissions under jamming. In 2020 IEEE Symposium on Computers and Communications (ISCC), pages 1–7. IEEE, (2020)
22. Citoni, Bruno, Ansari, Shuja, Abbasi, Qammer Hussain, Imran, Muhammad Ali, Hussain, Sajjad: Impact of inter-gateway distance on lorawan performance. Electronics **10**(18), 2197 (2021)
23. Mukhia, Raunak, Sarambage Jayarathna, Kalana Gayashan, Lertsinsrubtavee, Adisorn: Performance evaluation of lorawan forest fire monitoring network in the wild. In Proceedings of the 18th Asian Internet Engineering Conference, pages 96–104, (2023)
24. Safi, Abdullah, Ahmad, Zulfiqar, Jehangiri, Ali Imran, Latip, Rohaya, Zaman, Sardar Khaliq Uz., Khan, Muhammad Amir, Ghoniem, Rania M.: A fault tolerant surveillance system for fire detection and prevention using lorawan in smart buildings. Sensors **22**(21), 8411 (2022)
25. Elbsir, Houssem Eddin, Kassab, Mohammed, Bhiri, Sami, Bedoui, Mohamed Hedi: Evaluation of lorawan class b efficiency for downlink traffic. In 2020 16th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), pages 105–110, (2020)
26. Ilarizky, Alfian, Kurniawan, Azis, Subagyo, Erlangga Putro, Harwahyu, Ruki, Sari, Riri Fitri: Performance analysis of adaptive data rate scheme at network-server side in lorawan network. In 2021 2nd International Conference on ICT for Rural Development (IC-ICTRuDev), pages 1–5, (2021)
27. Chen, Mi, Mokdad, Lynda, Othman, Jalel Ben, Fourneau, Jean-Michel: Probabilistic model checking for unconfirmed transmission in lorawan on the mac layer. In GLOBECOM 2023 - 2023 IEEE Global Communications Conference, pages 2457–2462, (2023)
28. Martinez, Ivan, Nouvel, Fabienne, Lahoud, Samer, Tanguy, Philippe, Helou, Melhem El: On the performance evaluation of lorawan with re-transmissions under jamming. In 2020 IEEE Symposium on Computers and Communications (ISCC), pages 1–7, (2020)
29. Altayeb, Moez, Zennaro, Marco, Pietrosemoli, Ermanno, Manzoni, Pietro: Turbolora: Enhancing lorawan data rate via device synchronization. In 2021 IEEE 18th Annual Consumer Communications and Networking Conference (CCNC), pages 1–4, (2021)
30. Jain, Raj: The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling. John Wiley & Sons, (1990)
31. Silva, B., Matos, R., Callou, G., Figueiredo, J., Oliveira, D., Ferreira, J., Dantas, J., Lobo, A., Alves, V., Maciel, P.: Mercury: An integrated environment for performance and dependability evaluation of general systems. In Proceedings of the Industrial Track at 45th Dependable Systems and Networks Conference, DSN, pages 1–4, (2015)
32. Politis, Stavros N., Colombo, Paolo, Colombo, Gaia, Rekkas, Dimitrios M.: Design of experiments (doe) in pharmaceutical development. Drug Dev. Indust. Pharm. **43**(6), 889–901 (2017)
33. Martins, Isa, et al. FUKUDA (2018) Design of experiments (doe) applied to pharmaceutical and analytical quality by design (qbd). Science of the total environment, 54: e01006
34. Weissman, Steven A., Anderson, Neal G.: Design of experiments (doe) and process optimization. a review of recent publications. Org. Process Res. Dev. **19**, 1605–1633 (2015)
35. Lendrem, Dennis, Owen, Martin, Godbert, Sonya: Doe (design of experiments) in development chemistry: potential obstacles. Org. Process Res. Dev. **5**(3), 324–327 (2001)
36. Maciel, Paulo, Matos, Rubens, Silva, Bruno, Figueiredo, Jair, Oliveira, Danilo, Fé, Iure, Maciel, Ronierison, Dantas, Jamilson: Mercury: Performance and dependability evaluation of systems with exponential, expolynomial, and general distributions. In 2017 IEEE 22nd Pacific Rim international symposium on dependable computing (PRDC), pages 50–57. IEEE, (2017)

**Israel Araújo** Master's student in Computer Science at the Federal University of Piauí (UFPI) and a member of the Applied Research Group for Distributed Systems (PASID). Holds a bachelor's degree in Information Systems from the Federal University of Piauí (2016). Has experience in the fields of Computer Science and Information Systems, with an emphasis on system evaluation and distributed systems.

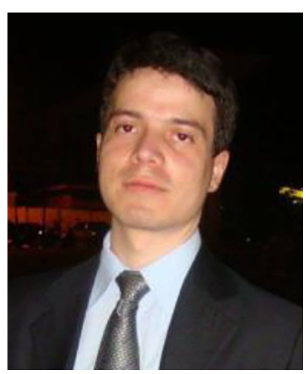
**Vandirleya Barbosa** Undergraduate student in Information Systems at the Federal University of Piauí (UFPI) and a member of the Applied Research Group for Distributed Systems (PASID). Her research interests focus on distributed systems, with an emphasis on availability and performance evaluation techniques. She is particularly interested in exploring methods to analyze and optimize system behavior in large-scale distributed environments, contributing to advancements in system reliability and efficiency.



**Luiz Nelson Lima** Undergraduate student in Information Systems at the Federal University of Piauí (UFPI) and a member of the Applied Research Group for Distributed Systems (PASID). He works in the areas of distributed computing systems, drones, and the Internet of Things (IoT), with an emphasis on performance evaluation, resource monitoring, and optimization in smart environments. His research focuses on the use of analytical models and Petri nets applied to dependability and disaster recovery in smart hospitals. Currently, he is dedicated to the development of hybrid systems integrating drones for environmental monitoring, logistics, and process optimization, leveraging fog, cloud, and IoT-based applications. He is particularly interested in projects that explore innovative solutions to improve efficiency, security, and sustainability in critical applications.

**Luis Guilherme Silva** Master's student at the Federal University of Piauí (UFPI), with a bachelor's degree in Information Systems earned in 2020 from the same institution. He has experience in the fields of Computer Science and Information Systems, focusing on system evaluation and distributed systems. His academic and research interests include distributed systems, the Internet of Things (IoT), and cloud computing, particularly in the design, implementation, and evaluation of scalable and reliable solutions. He is enthusiastic about exploring emerging technologies that enhance system performance, resource optimization, and integration in large-scale environments. Additionally, he seeks to contribute to innovative projects that address challenges in connectivity, interoperability, and efficient data management within intelligent and adaptive systems.

**Carlos Brito** Master's student at the Federal University of Piauí (UFPI), with a bachelor's degree in Information Systems earned in 2020 from the same institution. He has experience in the fields of Computer Science and Information Systems, focusing on system evaluation and distributed systems. His academic and research interests include distributed systems, the Internet of Things (IoT), and cloud computing, particularly in the design, implementation, and evaluation of scalable and reliable solutions. He is enthusiastic about exploring emerging technologies that enhance system performance, resource optimization, and integration in large-scale environments. Additionally, he seeks to contribute to innovative projects that address challenges in connectivity, interoperability, and efficient data management within intelligent and adaptive systems.

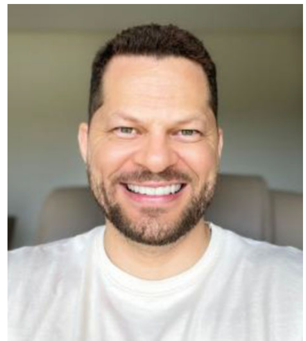


**Iure Fé** He holds a Bachelor's degree in Computer Science from the Federal University of Piauí (2009) and a Master's degree in Computer Science from the Federal University of Pernambuco (2017). Currently, he is pursuing a Ph.D. in the joint doctoral program between the Federal University of Maranhão (UFMA) and the Federal University of Piauí (UFPI). In addition, he works as a systems analyst at the 3rd Construction Engineering Battalion. His research focuses on the dependability and performance modeling of distributed systems. He has extensive experience in managing Linux environments, cloud architectures, and DevOps practices, with a strong emphasis on automation, scalability, and reliability in modern computing infrastructures.

**Lucas Lopes** Undergraduate student in Information Systems at the Federal University of Piauí (UFPI) and a member of the Applied Research Group for Distributed Systems (PASID). He works in the areas of distributed computing systems, drones, and the Internet of Things (IoT), with an emphasis on performance evaluation, resource monitoring, and optimization in smart environments. His research focuses on the use of analytical models and Petri nets applied to dependability and disaster recovery in smart hospitals. Currently, he is dedicated to the development of hybrid systems integrating drones for environmental monitoring, logistics, and process optimization, leveraging fog, cloud, and IoT-based applications. He is particularly interested in projects that explore innovative solutions to improve efficiency, security, and sustainability in critical applications.

**Ermeson Andrade** He holds a Bachelor's degree in Computer Science from the Catholic University of Pernambuco (2006), a Master's degree in Computer Science from the Federal University of Pernambuco (2009), a Ph.D. in Computer Science from the Federal University of Pernambuco (2014), and completed a post-doctoral fellowship at Vrije Universiteit Amsterdam (2016). Between 2010 and 2011, he participated in a sandwich Ph.D. program at Duke University in the United States. In 2022, he was awarded the prestigious Visiting Professor Fellowship from the Japan Society for the Promotion of Science (JSPS). Currently, he is an Associate Professor at the Federal Rural University of Pernambuco and a CNPq Productivity Research Fellow. He also co-leads the GMOS research group. His research interests include Dependability, Data Science, Modeling, and Performance Evaluation.

**Erico Leão** He holds a Bachelor's degree in Computer Science from the Federal University of Piauí (2005), a Master's degree from the Graduate Program in Electrical Engineering, with an emphasis on Computer Engineering, at the Federal University of Rio Grande do Norte (2007), and a Ph.D. in Informatics Engineering from the Faculty of Engineering of the University of Porto, Portugal (2016). Since 2008, he has been a faculty member in the Department of Computing (DC) at the Federal University of Piauí (UFPI). He has been part of the Graduate Program in Computer Science (PPGCC/UFPI) since 2017 and the joint Doctoral Program in Computer Science between UFPI and UFMA since 2020. His primary research and professional interests include large-scale wireless sensor networks, the Internet of Things, real-time communication protocols, and routing protocols.



**Francisco Airton Silva** He is currently an Assistant Professor at the Federal University of Piauí (UFPI), as well as a permanent member of the Master's program (PPGCC) and Doctoral program (DCCMAPI) at UFPI. He leads the Applied Research Group for Distributed Systems (PASID). He holds a Master's and Ph.D. from the Center for Informatics (CIn) at the Federal University of Pernambuco. He also has a specialization in Software Engineering from CEUT, Piauí, and a Bachelor's degree in Information Systems from the Federal Institute of Piauí. He has worked for major multinational companies such as OKI and EMC. During his Ph.D., he completed a one-year research internship at Sapienza University of Rome, Italy (2015). He is a reviewer for prominent scientific journals, including the *Journal of Reliable Intelligent Environments* and the *Cluster Computing Journal* by Springer. Since 2020, he has had at least one undergraduate student win an award in UFPI's internal scientific research competition every year. His research interests encompass performance evaluation and fault tolerance techniques in various contexts, such as Cloud Computing, the Internet of Things, and Vehicular Networks. Within these fields, he has produced dozens of impactful, high-quality publications, leading to his recognition as a CNPq Productivity Fellow in 2023.