Secure and Sustainable FUOTA in Constrained IoT Infrastructures

Maurizio Giacobbe[1,*,†], Salvatore Distefano[1,†]

[1]Department of Mathematics and Computer Sciences, Physical Sciences and Earth Sciences, Viale Ferdinando Stagno d'Alcontres, 31, Messina, 98166, Italy

Abstract

Low-power and long-range Internet of Things (IoT) devices and infrastructures, such as those based on Long Range Wide Area Network (LoRaWAN), are increasingly becoming integral components of Cyber-Physical Systems (CPS). Although they are advantageous in terms of economical and energy costs, Firmware Update Over-The-Air (FUOTA) is a challenge. The deployment of robust security mechanisms remains severely constrained by the limited energy, computational resources, and bandwidth of these devices. Stronger cryptographic primitives, larger keys, higher computational complexity, and increased communication overhead threaten the operational sustainability of constrained nodes and the efficiency of the underlying network.

This work proposes a performance evaluation method based on Generalized Stochastic Petri Nets (GSPN) to model and analyze the behavior of FUOTA procedures in resource-constrained IoT, based on LoRaWAN Class A infrastructure. Building on this analysis, we outline the technological and architectural evolutions required to enable secure and sustainable FUOTA mechanisms in such scenarios.

Keywords

Cybersecurity, Sustainability, Cyber-Physical Systems (CPS), Internet of Things (IoT), Firmware Update Over-The-Air (FUOTA), Generalized Stochastic Petri Net (GSPN).

1. Introduction

The Internet of Things (IoT) and Cyber-Physical Systems (CPS) are two significant concepts that have emerged from the advancement of digital technology. IoT refers to a network of physical devices equipped with sensors, software, and connectivity to autonomously collect and share data. While general IoT applications primarily focus on connectivity and data exchange, CPS combine computation, networking, and physical processes to activate a close link between physical components and their computational control, within a feedback system to monitor and manage processes in a real-world scenario. In this framework, IoT devices often serve as vital elements of a CPS infrastructure, collecting real-time data that computational algorithms use to drive and automate physical processes in complex areas, such as smart cities, healthcare, and advanced manufacturing. Therefore, the relationship between IoT and CPS is mutually beneficial thus enabling large-scale deployments of battery-powered sensor nodes, actuators, and embedded devices across heterogeneous contexts (e.g., industrial, agricultural, environmental, and urban applications).

Among different technologies supporting wide-area IoT connectivity, Low-Power Wide Area Networks (LPWAN), and in particular Long Range Wide Area Network (LoRaWAN), have emerged as strategic enablers for applications requiring long communication ranges, low deployment and operational costs, and multi-year device autonomy. Their ability to provide bidirectional connectivity over several kilometers, combined with minimal energy consumption, makes these technologies attractive for massive IoT deployments such as smart metering, precision farming, logistics monitoring, and environmental sensing.

Joint National Conference on Cybersecurity (ITASEC & SERICS 2026), February 09-13, 2026, Cagliari, IT

*Corresponding author.

†These authors contributed equally.

mgiacobbe@unime.it (M. Giacobbe); sdistefano@unime.it (S. Distefano)

0000-0001-6178-7132 (M. Giacobbe); 0000-0002-2752-626X (S. Distefano)

Despite these advantages, LPWAN technologies also impose severe limitations on the computational power, memory, communication bandwidth, and energy resources available to IoT end-devices. These constraints have significant implications for the design of security mechanisms, particularly when devices must support advanced functionalities such as remote software maintenance.

Firmware Update Over-The-Air (FUOTA) [1] plays a central role in ensuring long-term system reliability, vulnerability mitigation, and functional extensibility. A secure execution of FUOTA in constrained IoT environments remains an open challenge [2]. Ensuring the authenticity, integrity, and confidentiality of firmware images typically requires cryptographic mechanisms that introduce substantial computational overhead and communication-intensive operations, that impacts on the rigid energy and bandwidth budgets of LPWAN devices.

The situation becomes even more complex in light of emerging cryptographic requirements. For example, Post-Quantum Cryptography (PQC) [3] introduces novel primitives that often rely on mathematically harder problems, but also entail significantly larger key sizes, signatures, and cipher-texts compared to classical LoRaWAN. While these algorithms promise long-term security guarantees, their integration into constrained IoT systems is hindered by their computational and communication overheads. However, in the specific context of FUOTA, adopting advanced schemes in high-dense networks may lead to prohibitive increases in transmission time, energy consumption, and memory usage, thus jeopardizing device lifetime and network scalability.

Given these constraints, there is growing interest in quantitative methods capable of providing realistic assessments of the performance and sustainability of secure FUOTA in LPWAN environments [4]. Traditional analytical models and simulation-based approaches often capture only partial aspects of the system, such as transmission delays or energy expenditures, without considering the interplay between device behavior, firmware update, and network dynamics. Moreover, the stochastic nature of wireless communication (characterized by channel interference, packet collisions, duty cycle constraints, and retransmission policies) requires modeling frameworks that can accurately represent the probabilistic behavior and concurrent activities typical of low-power networks.

In such a context, Generalized Stochastic Petri Nets (GSPN) offer a powerful formalism to capture the temporal and probabilistic characteristics of distributed systems while preserving a high degree of modeling expressiveness. GSPN provide a structured way to represent concurrency, resource contention, stochastic delays, and system states, enabling a comprehensive evaluation of system-wide performance metrics. Their suitability for modeling communication protocols, embedded systems, and security procedures makes them a promising tool for analyzing secure FUOTA under realistic operational conditions. By integrating cryptographic operations, network transmission behaviors, and device-level resource constraints into a GSPN model, it is possible to quantitatively evaluate the impact of different security primitives on the overall performance and sustainability of the system.

By combining quantitative performance evaluation with architectural analysis, this work aims to provide an actionable guideline for designing next-generation FUOTA solutions.

The contributions of this paper are threefold. First, we identify the limitations and challenges inherent in securing FUOTA within LoRaWAN-based IoT systems. Second, we propose a comprehensive GSPN modeling framework capable of capturing the complex interactions between FUOTA, network behavior, and device constraints. Third, we outline a roadmap to enable secure and sustainable FUOTA through targeted architectural and protocol-level enhancements. Overall, these contributions advance the state-of-the-art in secure IoT maintenance and provide a foundation for future research in post-quantum readiness and long-term IoT resilience.

2. Preliminary Concepts

This Section provides the theoretical foundation and the technical background to understand the presented approach, with performance and reliability analysis. Establishing a rigorous framework is essential to characterize the complex interactions between the LoRaWAN communication protocol and the FUOTA procedure. First, we define the fundamental operational principles of LoRaWAN Class A.

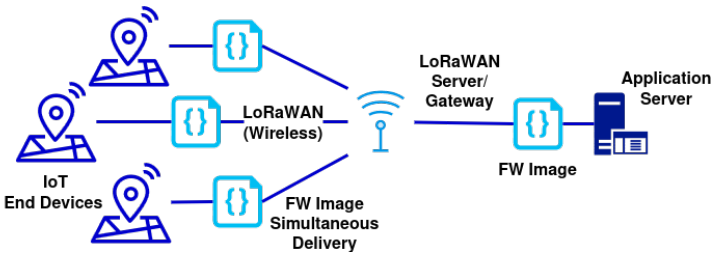

Figure 1: An overview of FUOTA in a LoRaWAN architecture.

Subsequently, we introduce the core metrics, thus allowing a quantitative assessment of the trade-offs between network performance, energy sustainability, and cybersecurity. Finally, we present related work.

2.1. Problem Description

2.1.1. LoRaWAN Device Classes and Multicast FUOTA Constraints

LoRaWAN defines three device operating classes: Class A, Class B, and Class C, which differ in terms of downlink availability, energy consumption, and network synchronization requirements. These differences have a direct and significant impact on the feasibility, duration, and scalability of FUOTA, particularly in duty-cycle–constrained sub-GHz bands such as *EU868 MHz ISM band*. The FUOTA framework is standardized by LoRa Alliance [5]; it relies on a combination of multicast communication and fragmentation mechanisms, namely the *Remote Multicast Setup* and the *Fragmented Data Block Transport*.

Table 1 presents a compact, protocol-oriented comparison of LoRaWAN device classes in the context of FUOTA, with emphasis on receive window behavior and multicast downlink delivery. The effectiveness of multicast-based FUOTA is strongly conditioned by the receive window model specified at the MAC layer for each LoRaWAN class. In Class A devices, multicast FUOTA downlinks can only be received during the RX1 and RX2 windows that follow an uplink transmission. Despite the use of multicast groups to reduce network load, the uplink-driven nature of receive window activation remains a fundamental limitation. As a result, multicast FUOTA in Class A typically requires periodic device-initiated uplinks to maintain synchronization with the multicast session, leading to extended update durations and limited scalability in duty-cycle–constrained regions.

Table 1
LoRaWAN Device Classes and FUOTA Framework Constraints

Characteristics	Class A	Class B	Class C
Multicast capability	Supported	Supported	Supported
Receive window	RX1/2 after TX	RX1/2 + RXbeacon	RX continuous
Uplink depend. for downlink	Required	Not required	Not required
Multicast transm. timing	Opportunistic	Beacon-aligned	Unconstrained
MAC-layer recept. constraint	Uplink-driven	Beacon-synchronized	Always-on reception
Duty-cycle impact	Severe	High	Moderate
Expected latency	Very high	High	Low
Expected energy consumption	Minimal	Moderate	High

Figure 1 shows an overview of the FUOTA in the LoRaWAN architecture. The application server requests the LoRaWAN Server/Gateway to deliver the firmware (FW) image to an end device or a group of end devices. This task is executed via the LoRaWAN wireless network according to the request.

In a typical LoRaWAN architecture, a large number of distributed end-devices communicate with collector nodes (i.e., gateways) using a star-of-stars topology. Network and application servers handle device management, data processing, and security functions. This architecture enables scalable

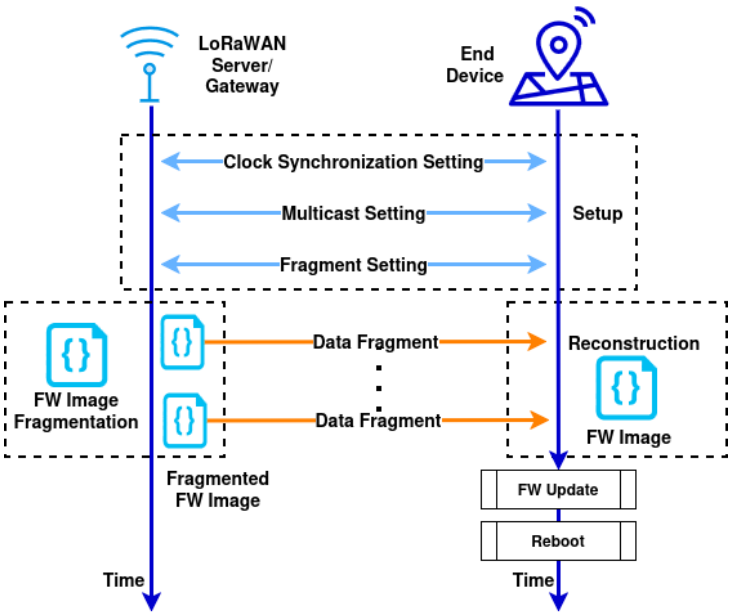

Figure 2: FUOTA Message Exchange between LoRaWAN Server/Gateway and End Device.

deployments but introduces limitations in terms of up-link and down-link capacity, strict duty-cycle constraints, and asymmetric communication patterns.

In this context, FUOTA requires the reliable delivery of fragmented firmware images over low-throughput links. Due to the typical limited payload size (bytes) and strict regulatory constraints on channel usage, firmware dissemination can span extended periods of time and involve complex coordination between devices, gateways, and backend services. Low-power IoT infrastructures can introduce significant delays and packet loss in dense networks.

Moreover, FUOTA procedures typically require devices to maintain specific listening windows that contrast with the device's need to conserve energy (continuity of service being secondary to battery life), thus increasing the risk of energy depletion before update completion.

2.1.2. Workflow

The sequence diagram in Figure 2 shows how the message exchange between the LoRaWAN Gateway (specifically the Network Server that is on the Gateway) and the end-device follows a rigorous operational sequence. At the beginning, firmware dissemination requires a setup phase that ensures the synchronization of fragment sizes and transport parameters between the server and the end device.

Therefore, the firmware image is delivered to the end-device as a sequence of data fragments. In our context, each fragment delivery is subject to the network's inherent PER during the down-link phase. Upon successful acquisition of all required fragments, the end-device reconstructs the fragmented data into a complete firmware image.

Finally, the device performs the internal firmware update using the reassembled image and executes a system reboot to apply the new code.

The above-mentioned characteristics make FUOTA in LoRaWAN ecosystems particularly sensitive to security and performance trade-offs. The *Software Updates for Internet of Things (SUIT) Working Group* of the *Internet Engineering Task Force (IETF)* recently carried out a standard architecture for firmware updates applicable to IoT devices [6], yet the implementation challenges in LPWAN environments remain significant, especially in terms of energy efficiency and battery lifetime, robust installation of firmware update so that it does not break the device functionality of the environment this device operates in, also taking into account the complexity of the decision making process of updating devices.

Device Components for FUOTA Implementation In the context of the LoRaWAN Class A FUOTA process analyzed in this work, the hardware and software components required on the end-device must be strictly optimized to cope with high latency and severe energy constraints. For a node based on the large diffused *Semtech SX127x* transceiver [7], the following architectural components are essential to manage the update process under performance limitations:

LPWAN-Optimized Transport Stack Firmware images for LoRa-based sensors typically range from 32 kB to 128 kB, thus the protocol stack must handle fragmentation and reassembly of minimal payloads (e.g., 15-byte data fragments). Crucially, in light of the high Packet Error Rate (PER) and the extended Mean Time To Update (MTTU) identified in dense networks, the retrieval mechanism must offer robust error recovery and the ability to resume transfers after prolonged interruptions without re-initiating the entire download cycle.

Persistence and Flash Management Each device requires the capability to write the received firmware fragments to persistent storage (Flash memory). In LoRaWAN-based IoT systems, the writing process must be synchronized with the network's duty-cycle to prevent critical energy depletion during intensive I/O operations.

Security Manifest and Integrity Verification A *manifest* is a structured metadata file that accompanies a firmware image. A manifest parser is necessary to verify digital signatures or Message Authentication Codes (MAC). In a LoRaWAN context, this verification serves as a primary defense against the unauthorized firmware injection attacks discussed in our threat model.

Firmware Unpacking and Decryption The device must be equipped to unpack, decompress, or decrypt the received image. Due to possible high MNRA, and the consequent risk of battery exhaustion, these computationally intensive operations are often offloaded to low-power secure elements to minimize the active-mode duration of the MCU.

FUOTA Status Tracker A persistent state tracker is required to monitor the update progress across thousands of up-link/down-link cycles. It is crucial to assure that the device can correctly transition back to the "uplink-waiting" state, and resume the update state even after network-induced failures.

These features are typically integrated into the application-level FUOTA agent rather than a minimal boot-loader.

2.1.3. Parameters and Metrics

The main parameters of the above described system are defined in Table 2.

The main metrics (i.e., *Key Performance Indicators - KPIs*), selected to objectively and quantitatively assess the objectives of our study, are described below.

Packet Error Rate (PER) The *Packet Error Rate (PER)* is defined as the ratio of the number of incorrectly received data packets to the total number of received packets, with a packet considered incorrect if at least one bit is erroneous [8], [9].

$$\text{PER} = \frac{\text{Number of incorrectly received packets}}{\text{Total number of received packets}} \tag{1}$$

In the context of the LoRaWAN FUOTA process, PER is a critical vulnerability factor: an high PER directly necessitates retransmission of the lost firmware data blocks, leading to reduce *reliability*, i.e. to increase the probability of FUOTA campaign failure or devices entering an inconsistent state. Moreover, each failed transmission requires an extra radio activation, resulting in unnecessary battery power consumption.

Table 2
LoRaWAN System Parameters and Functional Definitions

Parameter	Acronym	Brief Definition
Spreading Factor	SF	Number of chirps per symbol; determines the trade-off between range and data rate.
Bandwidth	BW	Width of the frequency band allocated for transmission (fixed at 125 kHz).
Coding Rate	CR	Proportion of the data stream that carries non-redundant information for error correction.
Fragment Size	S_{frag}	The actual amount of firmware data (in bytes) carried in each down-link transmission, excluding protocol overhead.
Redundancy Fragments	N_{red}	Additional packets generated via erasure coding (FEC) to recover lost fragments without requiring retransmissions.
Cyclic Redundancy Check	CRC	Error-detection mechanism to ensure the integrity of the received radio packet.
Preamble Length	PL	Sequence of symbols used for receiver synchronization before the start of the frame.
Number of Nodes	N	Total population of active end-devices competing for the shared radio channel.
Time on Air	ToA	Duration of the physical radio transmission over the wireless medium.
Uplink Request Interval	T_{up}	The periodic time interval between uplink transmissions that triggers the Class A receive windows for firmware fragment delivery.
Downlink Queue Capacity	Q_{max}	The maximum number of pending firmware fragments stored at the backend, awaiting the next available Class A receive window.
Backend Scheduling Latency	T_{proc}	The computational time required by the server to perform cryptographic signing, fragment encapsulation, and down-link scheduling.

Mean Time To Update (MTTU) The term *Mean Time To Update (MTTU)* [10] is utilized within this work as a specific performance metric for the FUOTA process in LoRaWAN Class A networks. It is rigorously defined as the expected value of the time elapsed from the initiation of a single firmware block transfer until its successful reception and acknowledgment.

The MTTU is rigorously defined as the expected value of the temporal interval between the initiation of a firmware block transfer and its successful finalization. Formally, let t_{init} be the stochastic instant representing the start of the transmission process for a single firmware block, and let t_{ack} be the instant of successful reception and acknowledgment, the MTTU is defined as follows:

$$\text{MTTU} = E[t_{ack} - t_{init}] \tag{2}$$

By assuming the FUOTA process is characterized by a sequence of discrete attempts governed by a transmission interval $T_{int} = \lambda_{T_1}^{-1}$ and a success probability $P_{success}$, the MTTU is analytically derived as:

$$\text{MTTU} = T_{int} \times \text{MNRA} = \frac{1}{\lambda_{T_1} \cdot P_{success}} \tag{3}$$

where *MNRA* represents the *Mean Number of Required Attempts*.

Mean Number of Required Attempts (MNRA) The *Mean Number of Required Attempts (MNRA)* metric [11] quantifies the efficiency of the FUOTA transfer process. It is defined as the expected number of total transmission attempts (successful or failed) needed to successfully deliver a single firmware data block from the server to the end device. The MNRA metric provides a clear measure of the operational overhead. A high MNRA value directly correlates with significant energy expenditure (wasted radio activations) and is intrinsically linked to network vulnerability (PER).

These metrics are a critical security indicator; as they determine the duration of the system's exposure to potential threats, minimizing the MTTU/MNRA is a primary objective to reduce the vulnerability window available for exploiting unpatched security weaknesses. Assuming the delivery of a critical security patch (i.e., the firmware block), the system is exposed to risk until all devices have successfully applied the update. The MTTU value directly determines the duration of this exposure period. A high MTTU (e.g., in high-density network scenarios) implies a prolonged vulnerability window, providing attackers with an extended time-frame to exploit the unpatched weakness.

Total Campaign Update Time (TCUT) While the MTTU characterizes the performance at the single-block level, the *Total Campaign Update Time (TCUT)* metric represents the cumulative expected time required to successfully complete the entire FUOTA process for a single device, from the initiation of the first fragment transfer to the final reception of the complete firmware image. The TCUT is derived by scaling the MTTU by the total number of fragments (N_{tot}) required to represent the full firmware binary. Formally, it is defined as:

$$\text{TCUT} = \text{MTTU} \times N_{tot} = \frac{N_{tot}}{\lambda_{T_1} \cdot P_{success}} \tag{4}$$

where N_{tot} is determined by the ratio between the total firmware size (S_{fw}) and the effective fragment size (S_{frag}):

$$N_{tot} = \left\lceil \frac{S_{fw}}{S_{frag}} \right\rceil \tag{5}$$

The TCUT provides a quantitative measure of the actual time window during which a device remains in an unpatched state.

2.2. Related Work

FUOTA methodologies have become indispensable for efficient firmware distribution and patching by remote, including timely responses to security vulnerabilities.

The main firmware update mechanisms for IoT, and in particular the ones about FUOTA are presented in [12]. Most of the low-power IoT devices do not have enough capabilities to securely conduct the FUOTA process; this is the most blocking challenge for the integrity and authenticity checks. The authors present a state-of-the-art of the currently proposed solutions, paying attention on the standardization and well-known industrial solutions.

A survey on the *over-the-air programming (OTAP)* techniques, which can be applied to IoT networks, is presented in [13].

Secure and lightweight FUOTA Mechanism for the IoT are investigated in [14]. The study proposes a FUOTA method designed to counter *man-in-the-middle* attacks within such constrained environments.

Controlling the energy consumption and update delivery latency of FUOTA are key challenges; on this end, a transmission scheme that provides fine-grain control over the energy-latency tradeoff in a LoRaWAN infrastructure [15].

A fundamental requirement for secure FUOTA is the establishment of *trustworthiness* between the architectural involved entities. Trustworthiness denotes the ability of a system or entity to provide verifiable assurance that it will operate as expected, even in the presence of faults, malicious actions, or adverse conditions.

In constrained IoT environments, trustworthiness is challenged not just by potential attackers, but by network unreliability. For example, a device cannot reliably authenticate the source of an update if the network link is unstable or saturated. The operational feasibility of robust authentication mechanisms based on static credentials is put at risk by network congestion and potential Denial-of-Service (DoS) and Distributed DoS (DDoS) attacks [16].

Remote attestation [17] is a fundamental requirement for secure FUOTA, enabling a device to prove its state. Implementing remote attestation in LoRaWAN-based IoT systems is particularly difficult due to performance constraints, i.e.: (i) Attestation protocols often rely on cryptographic operations and message exchanges that introduce significant computational and communication overhead; (ii) limited bandwidth, high latency, and vulnerability to packet loss of LPWAN links severely restrict the size and frequency of attestation messages; (iii) unsuccessful update, or natural network congestion, results in substantial energy depletion, thus rendering devices unavailable and insecure.

Our analysis shows that network conditions, rather than cryptographic complexity, often become the primary bottleneck for establishing timely bidirectional assurance, which is a cornerstone of trustworthy update mechanisms.

3. Proposed Approach

This Section presents a GSPN model designed to analyze the "trade-offs" performance and reliability of a FUOTA process within a LoRaWAN Class A architecture described in Section 2.1. The model specifically targets the vulnerability associated with the limited reception windows, and potential radio interference during data block transmission.

3.1. Generalized Stochastic Petri Nets

Generalized Stochastic Petri Nets (GSPN) provide a flexible and expressive formalism for modeling discrete–event systems that exhibit both logical interactions and stochastic timing. Compared to other graphical modeling techniques, GSPN naturally capture the interplay between concurrent activities, resource contention, delays, and probabilistic behaviors. Such characteristics are essential when assessing how current systems, often optimized around classical cryptography with limited computational and memory demands, react to the increased complexity and timing variability introduced by more advanced algorithms.

A GSPN is composed of *places*, *transitions*, *arcs*, and *tokens*. Tokens reside in places and move between them according to the firing of transitions, generating different system configurations known as *markings*. The graphical structure is a bipartite directed graph, where places and transitions form the two node classes. A marked GSPN is defined as a tuple (P, T, I, O, M) where P is the set of places, T the set of transitions, I and O the input and output relations, and M the marking that assigns a number of tokens to each place.

The execution semantics of a GSPN extend those of classical Petri Nets by introducing two types of transitions: (i) *Immediate transitions*, which fire as soon as they are enabled and capture instantaneous logical behavior; (ii) *Timed transitions*, whose firing delay follows an exponential distribution. This enables performance and reliability evaluation, particularly relevant when estimating the overhead of integrating post-quantum schemes into existing processes.

A transition is enabled when all its input places contain the required number of tokens and, if present, inhibitor arcs indicate that specific places must be empty. Conditioning functions may further refine the enabling conditions, allowing compact representation of complex logical constraints without increasing the modeling power.

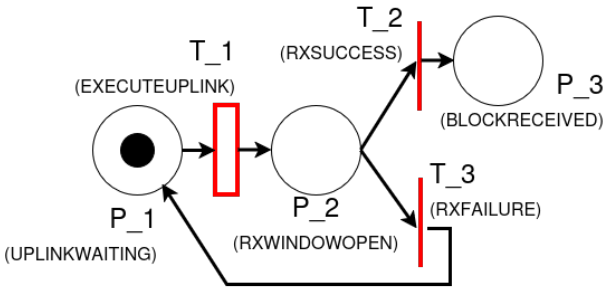

Figure 3: GSPN Model representing the FUOTA process.

Conflicts among enabled transitions are resolved according to well-defined execution policies. For timed transitions, the *race policy* determines which event occurs first, making the timing behavior fully stochastic. Immediate transitions, on the other hand, follow the *preselection policy*, ensuring that all instantaneous events are resolved before timed transitions are considered.

Given an initial marking, the evolution of the system can be described through its *reachability set* and the corresponding *reachability graph*. These structures allow systematic exploration of all possible system states and constitute the basis for evaluating how architectural changes (e.g., those required to support post-quantum defenses) affect performance, resource usage, and system resilience.

In this work, the GSPN formalism models and analyzes the dynamic behavior of a standard LoRaWAN Class A infrastructure to update the firmware for an end device. This approach enables a quantitative assessment of network vulnerabilities and operational limitations that constrain the application of essential defensive technologies, such as timely security patching.

3.2. GSPN Model

The GSPN model abstracts the communication flow for a single firmware data block transfer. The model represented in Figure 3 comprises three places (P) representing system states and three transitions (T) representing events, detailed in Table 3.

Table 3
Characterization of the GSPN Model representing the FUOTA process.

ID	Name	Type	Rate/Weight (Parameter)
P_1	*UplinkWaiting*	Place	N Initial Tokens
P_2	*RXWindowOpen*	Place	0 Initial Tokens
P_3	*BlockReceived*	Place	0 Initial Tokens (Absorbing State)
T_1	*ExecuteUplink*	Exponential	$\lambda_{T_1} = 1/T_{up}$
T_2	*RXSuccess*	Immediate	Normalized Weight $W_{T_2} \in [0, 1]$
T_3	*RXFailure*	Immediate	Normalized Weight $W_{T_3} \in [0, 1]$

The model operates as follows:

1. Initially, a token resides in P_1 (*UplinkWaiting*), representing an idle device awaiting its periodic communication interval.
2. Transition T_1 (*ExecuteUplink*) fires stochastically with rate λ_{T_1}, moving the token to P_2 (*RXWindowOpen*). This models the device waking up, sending an uplink, and opening its brief reception window.
3. From P_2, an immediate competition occurs between T_2 (*RXSuccess*) and T_3 (*RXFailure*). The outcome is governed by their respective weights, modeling the probability of successful reception versus failure due to collisions.
4. A firing of T_2 signifies success, moving the token to P_3, an absorbing state indicating successful block transfer.

5. A firing of T_3 (failure) returns the token to P_1, modeling the need for a subsequent uplink cycle and a reattempt, thereby capturing the energy and time waste associated with the vulnerability.

4. Case Study

In this Section, we present a case study which extends a preliminary work [18] concerning a LoRaWAN-based IoT architecture for wildfire monitoring in rural and forested environments. More specifically, we consider the system parameters reported in Table 4 that are consistent with the above scenario.

Table 4
LoRaWAN Configuration Parameters

Parameter	Acronym	Case Study Value
Spreading Factor	SF	10
Bandwidth	BW	125 kHz
Coding Rate	CR	4/5
Fragment Size	S_{frag}	15 Bytes
Redundancy Fragments	N_{red}	N/A (FEC not applied in base model)
Cyclic Redundancy Check	CRC	Enabled
Preamble Length	PL	8 Symbols
Number of Nodes	N	10, 30, 60, 120, 240
Time on Air	ToA	412 ms
Uplink Request Interval	T_{up}	60 s
Downlink Queue Capacity	Q_{max}	500 Packets
Backend Scheduling Latency	T_{proc}	100 ms

4.1. Stochastic Performance Evaluation and Numerical Results

The goal of the previous work was to analyze the capacity planning of gateways. Although datasheets of most common LoRaWAN gateways expose a capacity to manage a network of thousands of end devices/nodes, this indication results to be not realistic in constrain scenarios (e.g., in forests, dense urban centers, or high disturbed environments).

In this study, we extend our consideration to the impact of FUOTA, expecting for a further reduction of the number of end-nodes that can be managed in a secure and sustainable manner by each LoRaWAN gateway.

The 60 seconds T_{up} is selected as a design trade-off between timely environmental data collection and efficient network resource utilization. For example, in the above-mentioned critical scenario of wildfire prevention, frequent monitoring of key environmental parameters such as temperature, humidity, and air quality is crucial for the early detection of abnormal patterns that may indicate potential fire hazards.

The GSPN model was analyzed using the techniques implemented within the **WebSPN** tool [19] to derive key performance and reliability metrics. The focus is on balancing the system's performance (time) and sustainability (energy efficiency) across varying network densities as defined by the number of active nodes (10, 30, 60, 120, and 240 nodes). Due to potential collisions, the PER metric acts as the primary vulnerability factor.

Packet Error Rate (PER). PER is calculated to define the weights of the immediate transitions (T_2 and T_3). It models the intrinsic vulnerability of the radio link (primarily collisions in a pure Aloha system). This rate directly defines the competition probability between the success and failure transitions:

$$P_{\text{failure}} = \text{PER} = \frac{W_{T_3}}{W_{T_2} + W_{T_3}} \tag{6}$$

The GSPN analysis then uses this probability to stochastically determine the path a token takes from P_2 to either P_1 (failure loop) or P_3 (success).

The specific PER values used as input parameters in the GSPN model were derived analytically using the principles of a pure Aloha channel, which models the behavior of a Class A LoRaWAN uplink channel.

In a pure Aloha system, a packet is successfully received only if it does not collide with any other packet during its transmission duration. A collision occurs if another node starts transmitting within a vulnerable window that spans twice the packet's ToA. The probability of a successful transmission (P_{success}) is given by the following equation, where G is the normalized offered load of the network:

$$P_{\text{success}} = e^{-2G} \tag{7}$$

As a consequence:

$$\text{PER} = 1 - P_{\text{success}} = 1 - e^{-2G} \tag{8}$$

The load G is calculated based on the network configuration parameters provided in Table 4, specifically N, ToA, and T_{up}, as shown below:

$$G = N \times \frac{\text{ToA}}{T_{\text{up}}} \tag{9}$$

$$G = N \times \frac{0.412 \text{ s}}{60 \text{ s}} \approx N \times 0.00687 \tag{10}$$

By substituting the number of nodes (N) for each scenario into the formulas above, we derive the PER used to parameterize the GSPN weights W_{T_2} and W_{T_3}:

Table 5
Analytical Derivation of Network Load (G) and PER

Nodes (N)	**Offered Load (G)**	**Formula ($1 - e^{-2G}$)**	**Resulting PER**
10	0.0687	$1 - e^{-0.1374}$	12.8%
30	0.2060	$1 - e^{-0.4120}$	33.7%
60	0.4120	$1 - e^{-0.8240}$	56.0%
120	0.8240	$1 - e^{-1.6480}$	80.8%
240	1.6480	$1 - e^{-3.2960}$	96.3%

These analytically derived percentages confirm that the network performance degrades rapidly with increased load, justifying the input weights used in the GSPN model for the *RXFailure* transition (T_3).

Table 6
Normalized Immediate Transition Weights per Node Scenario

Nodes (N)	**Weight W_{T_2} (Success)**	**Weight W_{T_3} (Failure)**
10	0.872	0.128
30	0.663	0.337
60	0.440	0.560
120	0.192	0.808
240	0.037	0.963

Table 6 illustrates the normalized weights for the GSPN immediate transitions.

The data reveals a critical inverse relationship between the number of active nodes and the success probability (W_{T_2}). As network density increases, the weight of the *RXFailure* transition (T_3) prevails. This shift precisely quantifies the network's vulnerability: moving from a 12.8% failure probability (10 nodes) to a 96.3% failure probability (240 nodes) fundamentally changes the model's behavior, and therefore leading to drastically increased the MTTU and the energy waste across the various scenarios.

MTTU and MNRA. Table 7 summarizes the results of the comparative GSPN analysis, showing how the density of the LoRaWAN network affects the performance and sustainability of a single Class A FUOTA. The quantitative results, that are derived from the GSPN model analysis, directly correspond to the behavior of tokens moving through the places and transitions of the GSPN model under varying network loads.

Table 7
Comparative Results of GSPN Analysis Across Network Densities (hh indicates hours, dd indicates days).

Metric	10 Nodes	30 Nodes	60 Nodes	120 Nodes	240 Nodes
PER (Vulnerability)	12.8%	33.7%	56.0%	80.8%	96.3%
MTTU per Block	68.8 s	90.5 s	136.4 s	312.5 s	1621.6 s
MNRA	1.15	1.51	2.27	5.21	27.03
TCUT (5 kB Patch)	**6.54 hh**	**8.60 hh**	**12.96 hh**	**29.69 hh**	**154.05 hh**
TCUT (30 kB FW)	**1.63 dd**	**2.15 dd**	**3.24 dd**	**7.42 dd**	**38.56 dd**

The input PER parameter determines the probability split in the immediate competition between T_2 (*RXSuccess*) and T_3 (*RXFailure*) when a token resides in P_2 (*RXWindowOpen*). A high PER (e.g., 96.3% for 240 nodes) means the weight of T_3 is significantly higher than T_2. Consequently, the token is highly likely to fire T_3, which cycles it back to P_1 (*UplinkWaiting*). This represents the failure loop, and it highlights the severity of the network's vulnerability in these dense scenarios.

The MTTU (P_1-P_3) is related to the cumulative time spent in the timed place P_1. This time is stochastic and governed by the rate $\lambda_{T_1} = 1/60\ \text{s}^{-1}$. The total completion time is the sum of all time spent in P_1 across all necessary attempts until P_3 is reached. As PER increases, more failure loops occur (more visits to P_1), causing the MTTU to escalate from 68.8s to 1621.6s. This shows the model correctly predicting performance degradation based on the increased firing rate of the T_3 transition.

The MNRA is analytically derived from the GSPN model parameters: the probability of a single-attempt success (P_{success}). This probability is determined by the normalized weights of the immediate transitions T_2 (*RXSuccess*) and T_3 (*RXFailure*). Therefore, the MNRA is the inverse of P_{success}, forming an expected value based on a geometric distribution:

$$\text{MNRA} = \frac{1}{P_{\text{success}}} = \frac{1}{W_{T_2}} \tag{11}$$

This metric directly corresponds to the average number of times the token must traverse the GSPN cycle ($P_1 - T_1 - P_2 - T_3 - P_1$) before reaching the absorbing state P_3.

Figure 4 shows a strong direct correlation between the network's vulnerability (PER on the right axis) and the FUOTA performance (MTTU on the left axis). The increase in PER drives the GSPN model to cycle through the failure loop (T_3 back to P_1) more frequently. This increased looping directly results in the escalation of the MTTU metric. The visual alignment of the sharp increases in both curves confirms that the primary driver of poor FUOTA performance in dense networks is the fundamental vulnerability of the shared radio channel.

4.2. Sustainability and Cybersecurity Implications

The scaled results underscore a critical trade-off between network density, energy sustainability, and security. The **energy sustainability** of the FUOTA process is intrinsically linked to the *MNRA* metric. While a 30 kB firmware is relatively small, the operational overhead in high-density scenarios remains prohibitive. At 240 nodes, each successfully delivered fragment requires an average of 27.03 attempts. This exponential increase in radio activations leads to massive energy depletion, potentially exhausting the device's battery life before the update campaign reaches completion. In such congested environments, the energy footprint per successful byte becomes unsustainable for battery-powered Class A devices.

From a **cybersecurity** perspective, the *TCUT* serves as a quantitative measure of the **window of vulnerability**. If the FUOTA campaign aims to deliver a critical security patch, the device remains exposed to potential exploits until the final block is received and verified. The impact of network

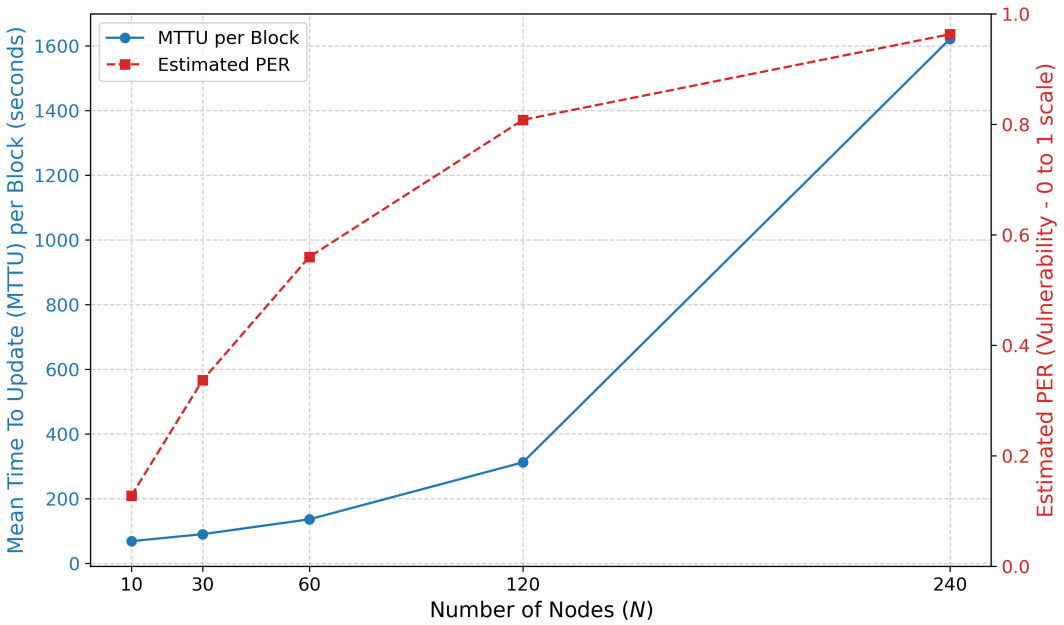

Figure 4: Comparative GSPN Analysis: Correlation Between Packet Error Rate (PER) and MTTU

congestion is evident even when considering a minimal 5 kB security patch ($N_{tot} = 342$): (i) in low-density scenarios (10-30 nodes), the vulnerability window for a 5 kB patch ranges from 6.54 to 8.60 hours, allowing for a relatively rapid incident response; (ii) in high-density scenarios (120-240 nodes), this window escalates to between 29.69 and 154.05 hours. When considering the full 30 kB firmware, the window extends from 7.4 to over 38 days in congested environments, that is operationally unacceptable for mission-critical infrastructures, as it provides adversaries with an extensive time-frame to exploit known unpatched weaknesses across the network.

5. Conclusion and Future Works

In this study, we proposed a performance evaluation method based on Generalized Stochastic Petri Net (GSPN) to model and analyze the behavior of Firmware Update Over-The-Air (FUOTA) procedures in resource-constrained IoT environments. More specifically, we presented a case study concerning FUOTA campaigns in LoRaWAN Class A for five different network load scenarios. Analytical findings show a drastic increase in the Packet Error Rate (PER) with network density (from 12.8% at 10 nodes to 96.3% at 240 nodes). This high vulnerability directly translates into increased energy overhead, quantified by the Mean Number of Required Attempts (MNRA), and significantly extended update durations, measured by the Mean Time To Update (MTTU) per block. The results highlight a critical trade-off: while Class A offers high energy efficiency for normal operations, its FUOTA mechanism is highly inefficient in congested environments. Cybersecurity practices (rapid patching) are inherently constrained by the LoRaWAN limitations, and the vulnerabilities quantified by the GSPN model. Effective FUOTA solutions in dense IoT networks necessitate the implementation of mitigation strategies, such as network segmentation or delta firmware, to ensure both reliability and sustainability.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] Kemsys, Understanding firmware over-the-air (fota), 2024. https://kemsys.com/blog/firmware-over-the-air/,lastaccessedDecember1,2025.

[2] V. Malumbres, J. Saldana, G. Berné, J. Modrego, Firmware updates over the air via lora: Unicast and broadcast combination for boosting update speed, Sensors 24 (2024). https://www.doi.org/10.3390/s24072104.

[3] NIST, Nist post-quantum cryptography project, 2017. https://csrc.nist.gov/projects/post-quantum-cryptography,lastaccessedDecember15,2025.

[4] B. P. Neves, A. Valente, V. D. N. Santos, Efficient runtime firmware update mechanism for lorawan class a devices, Eng 5 (2024) 2610–2632. https://www.doi.org/10.3390/eng5040137.

[5] ChirpStack Documentation, Fuota - firmware update over the air, https://www.chirpstack.io/docs/chirpstack/features/fuota.html, 2025. Last accessed December 17, 2025; ChirpStack open-source LoRaWAN® Network Server documentation.

[6] B. Moran, D. Brown, M. Meriac, H. Tschofenig, A firmware update architecture for internet of things, 2021. https://www.rfc-editor.org/rfc/rfc9019,lastaccessedDecember13,2025.

[7] Semtech, Sx1276 lora connect transceiver, https://www.semtech.com/products/wireless-rf/lora-connect/sx1276, 2025. Last accessed December 16, 2025.

[8] ScienceDirect Topics, Packet error rate, 2024. https://www.sciencedirect.com/topics/computer-science/packet-error-rate,lastaccessedDecember05,2025.

[9] Z. Ali, S. Henna, A. Akhunzada, M. Raza, S. Kim, Performance evaluation of lorawan for green internet of things, IEEE Access 7 (2019) 164102–164112. https://www.doi.org/10.1109/ACCESS.2019.2943720.

[10] M. Ajmone Marsan, G. Balbo, A. Bobbio, G. Chiola, G. Conte, A. Cumani, The effect of execution policies on the semantics and analysis of stochastic petri nets, IEEE Transactions on Software Engineering 15 (1989) 832–846. https:www.doi.org/10.1109/32.29483.

[11] M. J. Daniels, D. Jackson, W. Feng, I. R. White, Pattern mixture models for the analysis of repeated attempt designs, Biometrics 71 (2015) 1160–1167. https://doi.org/10.1111/biom.12353.

[12] S. El Jaouhari, E. Bouvet, Secure firmware over-the-air updates for iot: Survey, challenges, and discussions, Internet of Things 18 (2022) 100508. https://doi.org/10.1016/j.iot.2022.100508.

[13] K. Arakadakis, P. Charalampidis, A. Makrogiannakis, A. Fragkiadakis, Firmware over-the-air programming techniques for iot networks - a survey, ACM Comput. Surv. 54 (2021). https://doi.org/10.1145/3472292.

[14] C.-Y. Park, S.-J. Lee, I.-G. Lee, Secure and lightweight firmware over-the-air update mechanism for internet of things, Electronics 14 (2025). https://www.doi.org/10.3390/electronics14081583.

[15] S. S. Borkotoky, Balancing the energy consumption and latency of over-the-air firmware updates in lorawan, IEEE Transactions on Industrial Informatics 21 (2025) 7403–7411. https://www.doi.org/10.1109/TII.2025.3563539.

[16] M. Ali, Y. Saleem, S. Hina, G. A. Shah, Ddosvit: Iot ddos attack detection for fortifying firmware over-the-air (ota) updates using vision transformer, Internet of Things 30 (2025) 101527. https://doi.org/10.1016/j.iot.2025.101527.

[17] M. Jakobsson, Secure remote attestation, Cryptology ePrint Archive, Paper 2018/031, 2018. https://eprint.iacr.org/2018/031.

[18] M. Giacobbe, G. Tricomib, A. Puliafito, M. Scarpa, Addressing decentralized lorawan-based wildfire monitoring in forested environments, in: 2025 23rd International Symposium on Network Computing and Applications (NCA), 2025, pp. 149–156. https://doi.org/10.1109/NCA67271.2025.00034.

[19] Marco Scarpa, WebSPN - Web-accessible non Markovian Petri net tool, https://webspn.unime.it/, 2025. Last accessed December 16, 2025.