Article:
Alberti, A.M., Bonomo, E., Santos, R.H. et al. (2026) Applying NovaGenesis: A service-oriented, self-organizing, and programmable IoT architecture for LoRa and Wi-Fi-based environmental monitoring. Computer Communications, 248. 108423. ISSN: 0140-3664

https://doi.org/10.1016/j.comcom.2026.108423

Applying NovaGenesis: A Service-Oriented, Self-Organizing, and Programmable IoT Architecture for LoRa and Wi-Fi-Based Environmental Monitoring

Antonio M. Alberti[a], Epper Bonomo[b], Rodrigo H. Santos[b], Victor A. de J. Alberti[c], Marcelo E. Pellenz[d], Rodrigo da Rosa Righi[e]

[a]*School of Computer Science, University of Leeds, Sir William Henry Bragg Building, Woodhouse, Leeds LS2 9JT, UK. Phone: +44 0113 343 5430*
[b]*ICT Laboratory, INATEL, João de Camargo 510, Centro, Santa Rita do Sapucaí, CEP 37540-000, Minas Gerais, Brazil. Phone: +55 35 34719200*
[c]*Escola Técnica de Eletrônica (ETE) Francisco Moreira da Costa, CEP 37536-016, Av. Sinhá Moreira, 350 - Centro - Santa Rita do Sapucaí, Minas Gerais, Brazil. Phone: +55 35 3473-3600*
[d]*Graduate Program in Computer Science (PPGIa) - Pontifícia Universidade Católica do Paraná, R. Imac. Conceição, 1155 - Prado Velho, Curitiba - PR, 80215-901, Brazil. Phone: +55 41 3271-1669*
[e]*Programa Interdisciplinar de Pós-Graduação em Computação Aplicada Universidade do Vale do Rio dos Sinos, CEP: 93.022-000, São Leopoldo, Rio Grande do Sul, Brazil. Phone: +55 51 3590-8161*

Abstract

This work integrates NovaGenesis (NG), a clean-slate IoT architecture, with LoRa technology within low-power wide-area networks (LPWAN), extending previous efforts on NG connectivity with Wi-Fi. The research aims to update the embedded version of NG and develop devices for seamless LoRa and Wi-Fi IoT operation. It evaluates NG's performance on LoRa and Wi-Fi, focusing on throughput, delay, and packet loss. Despite LPWAN limitations, the results show that the NG deployment is feasible, validating its self-organizing IoT life cycle to maintain service continuity between an ESP-32 and a data client. Performance meets the needs of IoT applications in agribusiness, logistics, and smart monitoring. In addition, a 24-hour environmental monitoring experiment

Email addresses: `a.m.alberti@leeds.ac.uk` (Antonio M. Alberti), `epper.bonomo@mtel.inatel.br` (Epper Bonomo), `rodrigo.hilario@gea.inatel.br` (Rodrigo H. Santos), `victoralexandrealberti@gmail.com` (Victor A. de J. Alberti), `marcelo.pellenz@pucpr.br` (Marcelo E. Pellenz), `rrrighi@unisinos.br` (Rodrigo da Rosa Righi)

was conducted in Santa Rita do Sapucaí (SRS), Minas Gerais, Brazil, where a commercial weather station was modified to integrate NG, allowing accurate collection of temperature, humidity, atmospheric pressure, wind conditions, solar radiation and UV index. The results met expected diurnal patterns in SRS, proving the reliability and precision of the sensors and communication infrastructure. This solution overcomes common IETF IoT stack limitations in devices naming, information provenance, entities identification, programmability via digital twins, programmability, services and devices self-organization, and trust formation, offering a robust platform for varied IoT scenarios in LPWAN environments. These are the key benefits of applying NovaGenesis for LoRa and Wi-Fi-based environmental monitoring.
Keywords: Internet architecture, NovaGenesis, IoT, LoRa, Self-Organization, Programmability, Semantic Rich, Service Orchestration, Contract-Based Operation, Self-Verifying Naming, Gateway, Wi-Fi, Ethernet.

1. Introduction

Interest in IoT has led the IETF to create a dual-stack framework to integrate IoT with the Internet, maintaining core protocols. This IPv6-based dual stack is suitable for scenarios with ample resources, while 6LoWPAN [1] is introduced for low-power, limited-compute environments. However, the dual-stack approach has inherent limitations that hinder IoT evolution: (a) restricted naming and resolution capabilities, impeding self-organization; (b) absence of ID/Loc splitting, causing identity loss during device mobility; (c) lack of contract-based orchestration for IoT resources; and (d) limited programmability of physical devices via digital twins.

An alternate research direction focuses on "clean-slate" approaches: entirely new architectures designed from scratch, which integrate learnings from existing technologies and are prepared for future service developments. NovaGenesis (NG) represents one such clean-slate architecture [2, 3, 4, 5, 6, 7, 8]. Unlike the dual-stack model, NG addresses IoT's critical limitations with an integrated approach, offering robust solutions for the naming and identification, ID/Loc decoupling, digital twin support [9], service orientation, and contract-based self-organization.

While other alternative IoT architectures, such as the eXpressive Internet architecture (XIA[1])[10], recursive InterNetwork architecture (RINA[2]) [11], and named-data networking (NDN[3]) [12] attempt to address similar challenges, each fails in key areas. For example, NDN employs content-centric naming and separates identification from location; however, it does not inherently support digital twins or contract-based orchestration. On the other hand, XIA provides self-verifying names and versatile addressing, yet it lacks innate support for service orientation and self-organization. RINA emphasizes security, naming, and addressing, but lacks integration with digital twins or sophisticated service management mechanisms. In comparison, NovaGenesis comprehensively covers these aspects and offers a cohesive framework to address the shortcomings of current alternative IoT architectures. NovaGenesis views digital twins as one of the core foundations of its architecture, serving as a vital link between the physical and virtual worlds. They represent assets as named services for dynamic service composition, while still incorporating the named IoT content supported by ICN.

Our previous research demonstrated the feasibility of implementing NG using Wi-Fi networks [2]. This work showed significant improvements in areas such as entity naming (E1), identifier-location decoupling (E2), and device representation through digital twins (E3). It also validated NG's microservice-oriented architecture and contract-based self-organization (E4) while ensuring built-in name-based security and trust formation (E5). However, Wi-Fi-based solutions face restrictions in communication distance, rendering them inadequate for scenarios that require extensive coverage or extremely low power usage.

In order to expand the capabilities of NovaGenesis for far-reaching IoT appli-

[1]https://www.cs.cmu.edu/ xia/
[2]https://csr.bu.edu/rina/about.html
[3]https://named-data.net/

cations, this study integrates NovaGenesis with long-range (LoRa) technology [**?**]. LoRa presents several benefits for the IoT, such as low power consumption, extended communication range (up to 10 km in rural areas and 5 km in urban areas), scalability and economic efficiency [13]. Its simple star topology enhances energy efficiency and its ability to balance sensitivity and data rate ensures long battery life. These characteristics position LoRa as a suitable choice to augment NovaGenesis in IoT environments, including applications such as smart cities, precision agriculture, industrial monitoring, and intelligent networks for non-critical devices.

Therefore, this research pushes the boundaries of current technology by illustrating how a service-oriented, self-organizing, and programmable IoT architecture can be effectively used in long-distance IoT networks. Although the NDN proposal has been integrated with LoRa, it does not encompass the comprehensive features of NG, including advanced name resolution, digital twin support, and contract-based self-organization. By implementing NG with cost-effective LoRa ESP-32 devices, we offer an efficient, scalable, and economically viable solution that extends the application of clean-slate IoT beyond the limitations of Wi-Fi networks.

In this context, this study demonstrates the feasibility of adapting NovaGenesis for use with LoRa as a link layer, thereby supporting the progression of Future IoT (FIoT) [2].In other words, the main research question of this article is as follows: Is it possible to bring the benefits of NovaGenesis (unlimited naming, name resolution, ID/Loc splitting, contract-based orchestration, and programmability via digital twins) to LoRa with acceptable performance in terms of packet loss through a LoRa/Wi-Fi gateway? Consequently, our main contributions are the following.

1. The proposal of an innovative solution for LoRa connectivity using the NovaGenesis model through a LoRa/Wi-Fi gateway. This design addresses the IoT limitations mentioned above in areas such as naming, provenance, traceability, identification, location, representation (through digital twin),

flexibility, programmability, heterogeneity, self-organization, and trust formation among devices, middleware, and services.

2. A novel experimental evaluation methodology and meticulous analysis of the benefits of integrating NovaGenesis with LoRa and Wi-Fi for low-power, wide-area, and lossy networks.

To achieve these contributions, the following actions have been carried out. First, we expanded the NG architecture to interact with LoRa and Wi-Fi for IoT connectivity. Selection of a device that can support LoRa for long-distance communications together with NG-embedded software. Adaptation of the previous code developed for NG to run over Wi-Fi for this new scenario with LoRa, that is, to improve a service called the Embedded Proxy Gateway Service (EPGS) to run on ESP-32 LoRa devices. Development of an evaluation methodology and adjustment of NG parameters considering LoRa characteristics, which proved to impact the entire solution. To validate the new IoT service lifecycle, taking advantage of the self-organizing, programmable, and contract-based operation of NG. Lastly, we present a practical use case that is integrated with a commercial environmental monitoring station from Ativa Soluções Ltda. The aim is to substantiate the feasibility of the solution for prompt deployment within the market.

The remainder of this article is organized as follows. Section 1 presents related work on clean-slate IoT. Section 3 provides an overview of the NovaGenesis convergent information architecture and its embedded IoT service. In Section 4, we demonstrate a proof-of-concept scenario to extend advances in NG IoT to a low-power and unreliable LoRa network. In addition, we describe the experimental methodology used to test our hypothesis. Section 5 discusses the experimental results obtained. In order to substantiate the practicability of integrating the solution with a commercial counterpart, Section presents an environmental monitoring use case. Section 7 provides suggestions to minimize losses in the LoRa and Wi-Fi networks. Finally, Section 8 concludes the paper.

2. Related Work

This section provides a review of related work on the convergence of IoT and clean-slate architectures. We searched the literature for articles that overlap these two subjects. Our search included the IEEE, Elsevier, MDPI, and Springer databases. The following keywords were interchanged when selecting articles to be reviewed: "future Internet", "FIA", "IoT", "LoRa", "ICN", "NDN", "MobilityFirst", "RINA". The articles were selected based on their relevance to the research being conducted.

Da Silva et al. [14] have provided a survey on how clean-slate architectures address IoT challenges. The discussion covers the following architectures: (i) Named-Data Networking (NDN) [15], eXpressive Internet Architecture (XIA) [10], and Recursive InterNetwork Architecture (RINA) [11]. The authors presented an in-depth discussion of how each of these architectures addresses IoT issues. The paper proposed XIA for IoT for the first time in the literature. The work covered a containerized IoT experimentation and evaluation tool. The tool was applied to evaluate the current dual IoT stack with IPv6 and the XIA approach. The experimental results demonstrated the feasibility and effectiveness of this tool in experimenting with an IPv6-XIA gateway for IoT. XIA does not adopt a service-oriented (E4) approach for its IoT network services, i.e., service life cycle is left for applications. There is no contract-based operation between IoT devices and client applications. However, XIA offers Self-Verifying Naming (SVN), name resolution, XIA identifiers (XIDs), and directed acyclic graph addresses (E1). XIA decouples the IDs from the locators (E2). However, there is no native support for digital twins (E4). XIA offers intrinsic security based on SVNs and other traditional security mechanisms (E5).

NDN has been extended to support some of the demands of IoT. A novel communication model (push-pull) was proposed to allow IoT devices to push data to network caches. This approach contrasts with the original NDN solution in which interest packets query IoT content. Shang et al. [16] have proposed an NDN stack for RIOT OS and IEEE 802.15.4. A demo application has been

Table 1: An empty circle ○ indicates an enabler is not employed, a complete circle ● shows it is fully considered, a half-filled circle ◐ means partial treatment, and N/A signifies not available.

				Future IoT Enablers				
Ref.	**Main Topic**	**IoT**	**Evaluation**	E1 - Naming, identification, and addressing	E2 - Identifier and location splitting	E3 - Device representation via digital twins	E4 - Service orientation and contract-based	E5 - Security, privacy, trust, etc.
[14]	XIA and IPv6 for IoT in Cooja and Docker	802.15.4	Mixed	●	●	○	○	●
[16]	NDN stack for RIOT OS and IEEE 802.15.4	802.15.4	Experiment	◐	●	○	○	●
[17]	NDN over LoRa	LoRa	Experiment	●	◐	○	○	◐
[18]	RINA security and IoT	N/A	N/A	●	●	○	○	◐
[19]	Lightweight version of RINA called *rlite*	N/A	N/A	●	●	○	○	○
[20]	NDN naming and forwarding scheme for smart campus	Wi-Fi	Simulation	●	●	◐	○	◐
[21]	Secure routing within NDN-based smart city	N/A	Simulation	◐	●	○	○	●
[22]	LoRa-ICN: IoT combined with NDN.	LoRa	Simulation	◐	●	○	○	◐
[23]	Improved LoRa-ICN.	LoRa	Mixed	◐	●	○	○	◐
[24]	NDNoT: mitigation of flooding attacks and μNDN protocol stack.	802.15.4	Mixed	◐	●	○	○	◐
[25]	Adaptive NDN-edge computing framework	N/A	Mixed	●	●	○	○	●
[26]	NDN priority-based popularity-aware caching	Wi-Fi	Simulation	●	●	○	○	○
[27]	NDN priority-based popularity-aware caching	Wi-Fi	Simulation	●	●	○	○	○
[28]	Networking and security enablers for industrial IoT	MQTT	Experiment	●	●	○	○	●
	This work	LoRa Wi-Fi	Experiment	●	●	●	●	◐

embedded on a 32-bit ARM Cortex device. NDN inherently provides a new naming for content (E1), decoupling data identification from its location (E2). However, a conspicuous deficiency lies in the absence of intrinsic device or asset representation through digital twins (E3); furthermore, native support for name resolution remains unaddressed. The paper does not discuss the adaptability, programmability, or self-organizing capabilities of IoT networks (E4), which are vital to developing responsive and scalable IoT systems. In addition, the paper elucidates the practicality of integrating the NDN data-centric communication and security model within resource-constrained IoT platforms (E5). The experimental performance was evaluated in terms of the use of the network cache, energy consumption, and wireless broadcast parameters.

NDN has been tested for the communication of LoRa devices without IP addresses in [17]. The work demonstrates the feasibility of combining LoRa and NDN to create a long-range wireless ad hoc network. The approach took advantage of NDN routing and security not only in the data plane but also in sending control messages in the control plane. Addresses and overcomes the challenges of developing a virtualized interface between the NDN Forwarding Daemon (NFD) and LoRa wireless transmission medium. The support for naming and name resolution depends on the official implementation of NDN (E1). Same for ID/Loc splitting (E2). The representation of the devices through digital twins is notably underrepresented in the paper (E3). The manuscript lacks explicit references to service-oriented paradigms or contract-based operational methodologies (E4). The paper provides moderate coverage of security-related aspects (E5). The experimental evaluation was conducted utilizing the Raspberry Pi 3 Model B as the primary computing platform.

Ramezanifarkhani et al. [18] have provided a survey on how RINA addresses IoT challenges. A detailed discussion on stacking, addressing (E1), security (E5), and attack mitigation of the RINA protocol has been provided. The authors address the support of RINA for IoT security (E5). RINA offers flexible naming and addressing (E1) and ID/Loc splitting (E2). Native support for digital twins as services is not explored in the paper (E3), as well as service orientation and contract-based operation (E4). There is no evidence to suggest that this has already been implemented within the RINA framework. The most substantial coverage is on the security and authentication aspects (E5). This strong emphasis reflects the primary aim of the paper: addressing the security challenges of IoT through RINA. The paper thoroughly discusses various security features of RINA, such as secure distributed interprocess facilities, hidden addresses, and authentication mechanisms, which contribute to enhancing the overall security posture of IoT networks. The paper mentions the investigation of RINA's operation on small, limited devices like wireless sensors in the RINAiSense project. However, no evaluation of any kind is provided.

A lightweight version of RINA called *rlite* is described in [19]. This version provides Wi-Fi connectivity for IoT scenarios. The Linux *hostapd* software implemented access point (AP) functions for registered IPC processes. The *rlite* repository[4] includes a detailed tutorial on how to start using RINA with Wi-Fi. RINA implements a fixed set of customizable virtual functions in every Inter-Process Communication (IPC) instance. These functions can be optimized for the IoT. The document does not address service orientation (E4), nor elaborate on the coverage of digital twins (E3).

In the realm of IoT, particularly for smart campus applications, Arshad et al. [20] presented a significant advance through their work on NDN with the introduction of a novel naming and forwarding scheme, NDN-NS and NDN-NFS, respectively. Their research addresses critical challenges such as naming, security, and flexibility, which are essential for the scalability and efficiency of IoT networks. Their methodology addresses critical issues such as loss of transparency and addresses corruption, while simultaneously ensuring robust identification and traceability through the implementation of hierarchical, attribute-based, and flat naming schemes (E1). The inclusion of self-verifying names demonstrated an innovative approach to ensuring data integrity and authenticity (E1 and E5). Moreover, consideration of mobility and content-centric networking paradigms highlights the adaptability of their solution in dynamic environments (E2). Inclusion of security measures, including authentication, authorization, and self-verifying names, further enhances the integrity and trust within the network (E5). In general, the NDN-NS and NDN-NFS frameworks introduced by Arshad et al. [20] present a scalable and robustly secure solution tailored for smart campus ecosystems based on IoT principles. The manuscript lacks focus on service orientation, device representation (E3) and self-organization (E4),

[4]https://github.com/rlite/rlite

which is natural, since NDN is not an approach focused on named-services and their dynamic composition. The results were obtained by simulating NDN over Wi-Fi.

Patil and Vydeki [21] presented an advance in the domain of secure routing within NDN-based IoT networks for smart city applications. The authors introduce the NATSR protocol, which integrates lightweight Elliptic Curve Cryptography (ECC) for node authentication and a trust-based mechanism for secure data forwarding (E5). Therefore, the study thoroughly addresses security, privacy, provenance, traceability, and trust (E5). However, the coverage of naming, name resolution, identification, and addressing is relatively weak (E1), with minimal attention to naming and named data. However, it is well established that NDN encompasses these aspects (E1), as well as ID/Loc splitting (E2). Furthermore, the concepts of flexibility, programmability, and self-organization (E4) are barely touched upon, and there is a complete absence of discussion on device or asset representation via digital twins (E3). The authors designed and implemented NDN routing methodologies using the NS3 simulator within an Ubuntu 14.04 environment, configured as a guest operating system on the VMware platform.

Kietzmann et al. [22] focused on integrating ICN principles with LoRa for IoT applications. The manuscript addresses the complexities inherent in accommodating NDN within LoRa technology and replaces the LoRaWAN Medium Access Control (MAC) layer with an IEEE 802.15.4 Deterministic and Synchronous Multi-Channel Extension (DSME). The researchers have established a simulation environment for LoRa-ICN, utilizing OMNeT++ in conjunction with the INET[5] framework. They have seamlessly integrated *ccnSim*[6] [29] to provide essential ICN support, alongside openDSME[7] for the facilitation of 802.15.4 DSME protocols. The simulation model leverages FLoRa [30], incorporating its wireless propagation model and physical layer (PHY) attributes. To the best of

[5]https://inet.omnetpp.org/
[6]https://nonsns.github.io/code/ccnSim/
[7]https://github.com/openDSME/openDSME

our knowledge, there is no support for name resolution in this setup (E1) and security (E5) on ccnSim. ID/Loc functionalities are inherently supported by NDN (E2). There is an absence of any discourse on digital twins (E3), as well as on service-oriented design or contract-based operational paradigms (E4). The enhanced iteration of this research can be accessed in [23]. This more recent work utilized a long-range producer application implemented on commercially available IoT hardware, specifically leveraging a Nordic nRF52840 microcontroller integrated with a Semtech SX 1276 LoRa transceiver. The system architecture utilizes an ICN-based Ethernet/LoRa gateway functioning as a coordinator for wireless nodes interfacing with a Linux-based workstation. This workstation acts as a bridge to a virtual TAP interface via an Ethernet connection.

Recent advances in Named Data Networking of Things (NDNoT) have focused predominantly on improving security vulnerabilities (E5), with a specific focus on interest flooding attacks, as described by Bilgili et al. [24]. Their work encompasses efficient data retrieval methodologies and the attenuation of the complexities inherent in data management within IoT ecosystems. The article introduces novel methodologies for the detection and mitigation of NDN packet flooding attacks of interest. The approach called IfNoT capitalized on the stateful architecture of NDN by continuously monitoring Pending Interest Tables (PITs) and dynamically adjusting the probability of forwarding interest packets based on the observed network behavior. As a limitation, the research does not explore the role of digital twins (E3) and does not address self-organization, service orientation, and contract-based operation (E4). Performance evaluation of the IfNoT mechanism on resource-constrained IoT devices was carried out within the Contiki-NG operating system and using the μ NDN protocol stack [31]. A simulated Cooja network environment was used, thereby affirming its viability for practical IoT applications due to Cooja's ability to emulate authentic IoT motes.

In [25], NDN was employed in a smart city context, providing adaptive caching, dynamic resource allocation, multi-objective optimization models, and edge computing to improve latency, energy efficiency, and throughput in heterogeneous IoT environments. Performance evaluations were based on the *NDNSim* 2.8 simulator integrated with *ns-3* and Docker-based virtualized edge nodes. The use of named data objects (E1) and interest packets in NDN highlights the transition from IP-based location-dependent addressing to content-centric naming and ID/LOC resolution (E2). However, the paper does not address digital twin concepts or the representation of physical assets for service-level exposition (E3). The paper prioritizes optimization and adaptive mechanisms, but does not delve into service-oriented network programmability (E4). The security of NDN-IoT systems, including PHY security integration, enhanced authentication, attribute-based encryption, access control policies, and mechanisms to mitigate cache poisoning and DDoS attacks are discussed.

Meng and Ahmad [26], developed and evaluated a Priority-based Content Popularity-Aware (PCPA) caching strategy to enhance the availability of NDN data, reduce latency, and optimize network resources. Performance evaluations were based on the *NDNSim* and focused on Wi-Fi. The paper addresses naming and identification challenges in constrained IoT environments (E1 and E2). Support for device or physical asset representation using digital twins is entirely absent (E3). Similarly, flexibility, programmability, and self-organization of IoT services were not mentioned (E4). Mechanisms or discussions related to broader security frameworks, provenance, or trust management are not provided (E5).

Rafique et al. [27] introduced SoftCaching, a software-defined ICN method that optimizes selection of caching nodes and computes optimal routing paths for named content. The aim was to optimize caching and routing in IoT service provisioning. The paper conducted simulations with the Python-based Icarus simulator and OpenNemMon to monitor the SDN per-flow traffic. Named content and name resolution were considered as implemented in the Icarus simulator (E1 and E2). However, as in most of the previously mentioned research, there is no coverage for device representation via digital twins (E3). The paper does not elaborate on the broader paradigms of self-organization, orchestration, or contract-based service management, limiting the scope of flexibility and service-centric programming (E4). The security and privacy challenges are acknowledged (E5). Network policies, such as access control lists, are used to enforce control over caching nodes and traffic flows. However, enhancing service security with contract-based operations or named services is unexplored.

In [28], Li et al. proposed a novel ICN framework to meet the evolving demands of industrial IoT, focusing on data localization, distributed caching, multicast communication, and decoupling of senders and receivers to ensure efficient content distribution. The work leveraged a publish-subscribe mechanism to provide scalable and efficient data delivery. It also developed a zero-knowledge proof protocol (ZKP) to authenticate user attributes and reduce unauthorized data requests (E5). The proposal covered naming scopes (E1), access policies tied to data objects (E5), and rendezvous nodes to manage namespaces and access controls. Rendezvous nodes act as a resolver for named data objects (NDOs) (E2) and as a temporary cache for NDOs. The paper integrates attribute-based encryption, blockchain and zero-knowledge proofs to improve security, privacy, and access control in industrial IoT (E5). The E3 and E4 enablers were not used, leading to limitations in device representation, named service-level orchestration, and self-organization.

The current state-of-the-art presented in Table 1 is predominantly focused on enablers E1, E2, and E5. Naturally, all ICN proposals (XIA, RINA, and NDN) address naming, name resolution, and in-network storage performance limitations existing in the current IETF stack for IoT, securing content by their names and providing access independently of localization. By changing the way content is named, accessed, temporarily cached, and delivered, ICN enables verification of the authenticity, integrity, and provenance of IoT data regardless of their location or transmission paths. However, practically all proposals for the use of ICN in IoT do not consider the importance of naming services, orchestrating them according to their secure names, and establishing contracts for a reliable dynamic composition of IoT services. The service level aspects are equally significant for the advancement of IoT as those related to ICN. However, as can be seen in Table 1, these aspects are overlooked. Built-in name-based security, self-organization, contract-based operation, improved flexibility, and trust formation of IoT services (E4) are largely neglected. Furthermore, the role of digital twins as a fundamental pillar of IoT architectures remains underexplored in related work. Virtual representations of physical devices by named services have the potential to improve asset monitoring, management, programmability, control, and automation. In addition, experimental tools and simulation environments, such as those used to evaluate IPv6-XIA gateways and NDN on various platforms, offer practical ways to validate theoretical advances and drive IoT innovation. However, only field tests can reliably prove the research hypotheses.

This work addresses the aforementioned literature gaps in IoT by experimentally evaluating the convergence of named services and content, decoupling IDs from locators, with in-network caching and name resolution, showcasing the proposal's potential through a representative environment monitoring use case. This work is among the few that integrates named content distribution and orchestration of IoT services with LoRa, a key IoT technology. In the subsequent section, we shall delineate the NovaGenesis architecture and elucidate how it comprehensively addresses these enabling factors (E1-E5) to resolve issues that remain unsolved by previous research. We will also demonstrate how the obtained experimental results substantiate the underlying concepts of the NovaGenesis approach.

3. NovaGenesis

This section outlines the NovaGenesis (NG) architecture, its key concepts, and implementation, highlighting how NG supports IoT, including the lifecycle of NG services in the IoT environment. It is applicable to any IoT technology, such as LoRa.

3.1. Concepts and Implementation

NovaGenesis reimagines the Internet by learning from its existing model. It comes from extensive research on the Future IoT requirements, technologies,

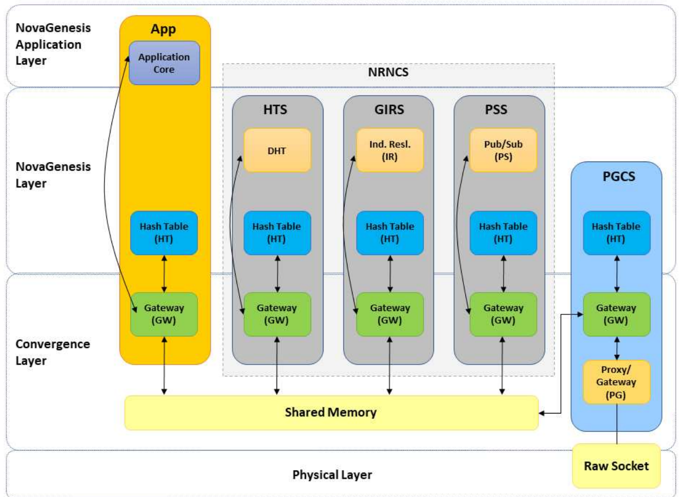

Figure 1: NovaGenesis protocol stack and core services in a domain.

and challenges, highlighting its flexibility in supporting emerging protocols-implemented-as-a-service (PIaaS) [5, 3] and addressing the service-centric design paradigm (E4). NG supports an information-centric approach, accessing and delivering information objects via self-verifying names (E1 and E5). It uses a network cache and a publish/subscribe model. NG employs a multi-strategy design to find synergies, allowing new protocols to be added as services (E4), including routing, content delivery, caching, software control, and fragmentation.

NovaGenesis integrates features into a single architecture [32, 6] for coexistence with legacy and emerging networks. It consists of three layers with distinct functions [2, 33, 3]: the *Convergence Layer* layer manages communication between internal NG services and adapts messages for external link layers like Ethernet, Wi-Fi, and LoRa; the *NovaGenesis Layer* layer houses specialized services; and the *Application Layer* supports user applications. Figure 1 depicts these layers.

The current NG implementation[8] comprises code blocks with clear functions for information exchange, processing, and storage. Shared memory (SHM)

[8]https://github.com/antonioalberti/novagenesis

serves as a communication bus within an OS, used by internal services to exchange NG messages. The gateway (GW) is an NG component that interacts directly with SHM, with an event-driven core for messaging and scheduling. SHM handles all service messages on the same OS, VM, or Docker. The hash table (HT) stores name bindings (NB) among names (E1 and E2), acting as a repository for local bindings and related information in NG services[6].

A hash table service (HTS) stores name bindings and their content in an HT data structure. For example, temperature measurements from a sensor can be stored as a file in the HTS folder on Linux. When processed by a *hash* function, this file generates a hash code or self-verifying name (SVN)(E1 and E5), which, combined with the file name, forms an NB.

A publish/subscribe service (PSS) enables interactions between publishers and subscribers regarding NB or content. It tracks preauthorized services for NB/content access using six primitives: (a) publish NB/data; (b) publish and notify; (c) subscribe to NB/data; (d) subscribe and notify; (e) deliver subscriptions to authorized clients; (f) revoke a published NB/data.

A generic indirection resolution service (GIRS) mediates between PSSes and HTSes, directing NBs/data from PSS to the appropriate HTS instance. When HTS and PSS combine to form the Name Resolution and Network Cache Service (NRNCS), GIRS becomes unnecessary, as the data is stored internally.

A proxy/gateway/controller service (PGCS) provides three key NG services: (a) a ***gateway*** for encapsulating and extracting NG messages over a link layer networking technology; (b) a ***proxy*** for representing physical devices, a feature known as digital twin (E3); and (c) a **software controller** for initiating the domain and configuring physical devices as needed. The connection between NG and software-defined networking (SDN) is discussed in [4]. PGCS can be seen not only as a software controller, but also as a digital twin for one or more IoT devices. It includes all the logic required to represent these devices at the service orchestration level, mirroring their capabilities and status. In addition, it can configure device functionalities as required by applications. A device can update its status by publishing bindings and associated content to PGCS.

Similarly, PGCS can update a device configuration by publishing NBs and files to an embedded EPGS (presented in Subsection 3.2.

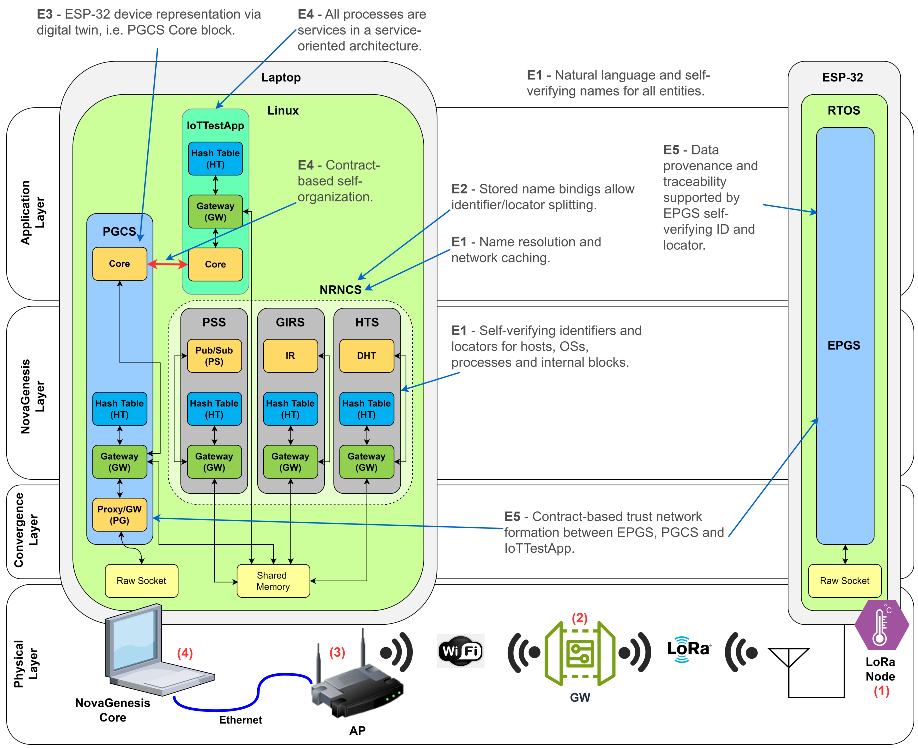

Figure 2: NovaGenesis IoT service (EPGS) interoperates with the NG core through its digital twin, PGCS. A LoRa Node (1) links to a LoRa/Wi-Fi Gateway (2), which connects to a Wi-Fi Access Point (3). The NG Core operates on a laptop (4), and EPGS runs on the LoRa Node (1). Temperature data flows from (1) to (4) using NovaGenesis messages transported over LoRa, Wi-Fi, and Ethernet across three links.

The final application (App) is a user-developed tool for data and command exchange with other applications and even devices. An example is “IoTTestApp”, which uses data from EPGS, an embedded PGCS version for IoT devices (Figure 2). IoTTestApp can establish a service contract directly with an IoT node via its digital twin (PGCS). A detailed description of this procedure is given in [2].

NG allows dynamic service composition by employing a distributed named-services life cycle, in which applications (e.g., IoTTestApp) can contract digital twins (PGCSes) to measure data according to required demands (E3 and E4). Before any publication or subscription of NBs and/or content on NRNCS, NG requires the construction of contracts between the services. This includes the IoTTestApp application, the PGCS digital twin, and the embedded EPGS on the IoT device. This orchestration is performed by the services themselves by directly publishing/subscribing on NRNCS without the help of brokers or buses. We can say that with NG, the orchestration of named services reaches IoT devices. In the next subsection, we will present how an IoT device can establish a service contract through its EPGS with its digital twin (PGCS) and then with a final targeted IoTTestApp.

3.2. Support for IoT — Fostering FIoT

From its inception in 2008, NG has uniquely supported IoT. EPGS [2] compacts NG's capabilities for devices with lightweight OSs like FreeRTOS and RTOS. They can be embedded in *hardware* such as Arduino, ESP-32, NXP, among other open platforms on the market. EPGS was developed as a shortened version of NG's PGCS, maintaining all features. Its operation is also guided by service contracts signed between publishing entities and subscribers through PSS (or NRNCS), in an architecture controlled by *software*.

EPGS follows the same NG principle of using SVNes generated from the *hardware* of IoT *nodes*, and is capable of generating, sending, receiving, and interpreting NG commands within predefined tasks for its life cycle. An NG command is a line of code in a message/packet that causes a function/action to be performed in a process (service) [4]. EPGS adapts messages for any link layer, fragments and reassembles packets, receives device information, sensors data, and optimizes algorithms to minimize ROM usage and energy consumption. EPGS[9] is an open source implemented for ESP-32.

NG service-oriented design (E3 and E4) covers the service's lifecycle from instantiation to release. In its embedded version, EPGS has implemented a

[9]https://github.com/Scalifax/EPGS_LORA_NODE

subset of the NG service lifecycle that contains the following steps [2]:

- **Hello** - A *Hello* message is sent on *broadcast* by PGCSes to IoT nodes, providing NG service components, interfaces, and IDs. Routing of *Hello* messages uses host-based host ID (HID), OSID, and process ID (PID). The *Hello* message includes PSS IDs for IoT data publishing and IoTTestApp subscriptions on Laptop (4). NovaGenesis thus implements entity naming (E1) and ID/Loc splitting (E2).

- **Exposition** - In a PSS/NRNCS domain, it publishes EPGS installation details. EPGS creates SVNes (HID, OSID, PID) to identify hardware via serial number, OS via RTOS naming, and process via memory position on RTOS, facilitating IoTTestApp's discovery and contracting of services via network cache information. The exposition message outlines device features, such as measurable variables, sensor accuracy, and operational range. NRNCS handles Name resolution (E1).

- **Discovery** - It allows IoT devices to find existing PGCSes (digital twins) and IoTTestApps in a domain by subscribing to keywords in natural language within the domain's PSS or NRNCS. EPGS checks if subscription data reveal new peer services at the core. Name bindings enable IoT service self-organization (E4).

- **Service Offering** - An IoTTestApp offers a Service Level Agreement (SLA) to an EPGS capable of sensing or actuating on the IoT side.

- **Service Acceptance** - EPGS accepts the IoTTestApp service offer, engages a device to deliver data to an application and establishes the NovaGenesis trust network (E5). In this way, the dynamic composition of services occurs using the names of the services, devices, and operating systems, both in the core and in the IoT part of the network.

- **Data Publication** - After establishing the SLA (a contract) between EPGS and IoTTestApp, EPGS publishes data to NRNCS in JSON format within an NG message payload. JSON is chosen for compatibility with IoTTestApp and offers efficient data handling. NG messages use a new scripting language developed in 2012 [5, 6].

- **Data Subscription** - IoTTestApp is notified of new IoT data in a domain cache (HTS or NRNCS). Applications subscribe to the data using SVNes, similar to MQTT, with integrity checked by self-verifying names (E5).

4. Applying NovaGenesis for LoRa Connectivity

Subsection 4.1 describes the convergent scenario analyzed, in which temperature readings are transmitted from an ESP-32 node using LoRa to a Wi-Fi gateway[10], which is connected to the NG core via Ethernet. Technical challenges include activating multithreading on the ESP-32, with the LoRa/Wi-Fi gateway facilitating real-time observation. The scenario also examines experimental conditions in São Paulo, Brazil, known for its crowded radio frequency, with the aim of evaluating performance under demanding circumstances. The setup utilizes Spreading Factor 7 (SF7) with a bandwidth of 500 KHz for a data rate of 21.9 kbps but currently lacks multi-access control, a feature targeted for future development. The experimental protocol presented in Subsection 4.2 documents the loss of messages in the NG using several files, including sampled temperature logs and Wireshark captures, with the data analyzed in MS Excel™ to monitor fragmentation and loss of messages in the network. Describes how these logs are matched to detect losses in the LoRa, Wi-Fi, and Ethernet network portions for a comprehensive performance analysis of the LoRa network's integration with NG. Tables 2 and 3 summarize the hardware and software used in this work, respectively.

4.1. Convergent Scenario

A convergent scenario was defined to expand the previous developments of the NG IoT to a low-power, lossy LoRa network. Figure 2 illustrates how

[10]https://github.com/Scalifax/LORA_WIFI_GATEWAY

LoRa is used to transport NG messages from a temperature sensor node (1) to a LoRa/Wi-Fi gateway (2). *Heltec* ESP-32 device has enough memory and processing capacity to ship EPGS, RTOS, and LoRa communication protocols together. This may seem like a straightforward task. However, it is actually quite intricate. This is not due to the complexity of NG, but rather because the firmware must be modified to support multithreading. Gateway (2) translates messages between LoRa and Wi-Fi. ESP-32 LoRa provided an open platform that supports custom development. It also allows for near-real-time monitoring, giving greater visibility in the analysis of integrated architectures. EPGS publishes measurements to IoTTestApp at the NG core's Application Layer.

Table 2: Hardware characteristics of the experimental setup.

Device	**Standards**	**Settings**	**Location**	**Additional Notes**
ESP-32 Node (Sensor)	LoRa (Heltec ESP-32) SX1276	SF7 spreading factor, 500 KHz bandwidth, 868/915MHz, ROM: 448K, SRAM: 520Kb, 21.9 kbps data rate, payload 222 bytes.	São Paulo, Brazil (high RF interference)	Firmware modified for multithreading; embeds EPGS, RTOS, LoRa protocols
LoRa/Wi-Fi Gateway	LoRa / Wi-Fi (802.11n)	2.4 GHz band, Channel 11	São Paulo, Brazil (crowded 2.4 GHz spectrum)	Translates messages between LoRa and Wi-Fi; supports near-real-time monitoring; no multi-access control on LoRa yet
Wi-Fi Access Point (AP)	Wi-Fi 802.11n	2.4 GHz, Channel 11	São Paulo, Brazil (subject to interference from cordless phones, Bluetooth, microwaves, etc.)	Connects Gateway to Ethernet network
Laptop running NG Core	Ethernet	Intel Core i5 processor, 2.67GHz, with 6GB of RAM. Ethernet link.	São Paulo, Brazil	Receives traffic from AP; runs NovaGenesis Core protocols
Aruba IAP-205RW	Wi-Fi 802.11n	2.4 GHz, Channel 11	São Paulo, Brazil	Commercial-grade AP from Aruba Networks; used to monitor RF interference levels at the experimental site.

Our experiments have been carried out in a region close to Paulista Avenue -

São Paulo - Brazil (Lat/Long 23º34’52”S 46º39’09”W). In the Wi-Fi portion of the network, which connects Gateway (2) to AP (3), the 802.11n standard was selected in the 2.4 GHz frequency band and with Channel 11. It is known that this spectrum band, as it is not licensed by regulatory agencies, is subject to interference due to the large number of devices operating in this band, such as cordless phones, *Bluetooth*, Microwaves, etc. This region is considered the worst RF interference point in Brazil. Our idea was to evaluate this proposal in one of the most challenging locations possible, with high radio frequency interference.

In the LoRa segment, the SF7 spreading factor was adopted with a bandwidth of 500 KHz, allowing a higher LoRa data transmission rate (21.9 kbps) with a *payload* of 222 bytes. As a limitation of this proof-of-concept (PoC), the LoRa communication channel does not support multi-access control. A future version of the gateway will provide this feature. The lack of this functionality is not considered critical at this time. Finally, the traffic from the AP (3) is transmitted over Ethernet to a laptop running NovaGenesis Core protocols (4).

4.2. Experimental Protocol

This article evaluates the performance of NG over LoRa by estimating the amount of field NG messages lost in the experimental scenario illustrated in Figure 2 and depicted in Figure 3. The IoTTestApp application that runs on the laptop generates a file “data.txt” during the data publication step of the life cycle presented previously in Subsection 3.2. This file stores the following information: (a) the timestamp related to the time a sample sent by the LoRa node is saved on the laptop; (b) the name of the IoT node that is publishing the measure; (c) the sequence number of the publication message forwarded to NRNCS; and (d) the instantaneous temperature measurement collected in Celsius degrees. It should be mentioned that the tests were performed with a two-second time interval between consecutive EPGS data publications. An example of this file is illustrated in Figure 4, with the letters marked in each column representing the information described above. In other words, this file contains statistics of the measurements received from EPGS in the NG IoTTestApp application.

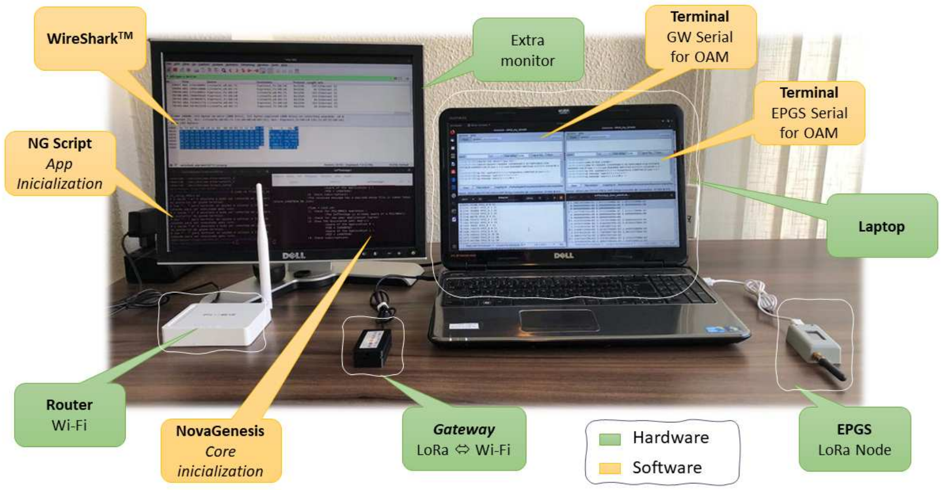

Figure 3: Depicts the same experimental scenario illustrated in Figure 2 with NG core and embedded NG in LoRa for temperature monitoring.

Wireshark^{TM} has also been installed on the laptop connected to the local Ethernet network with visibility of all packets transmitted to the NG core. Wireshark^{TM} was configured with a filter to display only NG messages that use "*Ethertype*=0x1234", therefore minimizing the number of records to be further verified and analyzed.

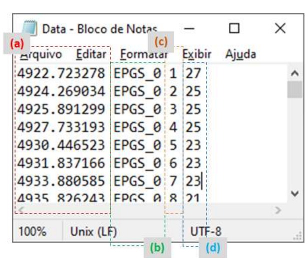

Figure 4: Partial print of the data.txt file created by the NG IoTTestApp. (a) timestamp; (b) node name; (c) sequence number; and (d) room temperature in Celsius.

Wireshark^{TM} *logs* have also been exported to a "wireshark.csv" file. To facilitate the search process for the desired performance information and the correlation with other files by means of key fields, this ".csv" file is imported into MS Excel^{TM}, which originates a new file formed by specific columns and

A	B	C	D	E	F	G	H	I	J	K	L
No	Time	Source	Destination	Protocol	Length	Info	Intervalo	Sample	Step/Fragment	Execution Time	Time Ratio
292	204,9420865	Espressi_fc:b8:c8	LiteonTe_e6:65:71	0x1234	222	Ethernet II	0,407777		Service Offering	1,844105051	
311	223,0679068	Espressi_fc:b8:c8	LiteonTe_e6:65:71	0x1234	103	Ethernet II	18,12582		Service Acceptance		
312	223,5849355	Espressi_fc:b8:c8	LiteonTe_e6:65:71	0x1234	222	Ethernet II	0,517029		Service Acceptance	0,517028672	
346	266,1793844	Espressi_fc:b8:c8	LiteonTe_e6:65:71	0x1234	198	Ethernet II	42,59445		Broadcast		
347	266,5876561	Espressi_fc:b8:c8	Broadcast	0x1234	222	Ethernet II	0,408272		Broadcast	0,4082717	
348	266,9981343	Espressi_fc:b8:c8	Broadcast	0x1234	92	Ethernet II	0,410478	0	Data Publishing - Frag1 Temperature		83,86701095
349	267,5187823	Espressi_fc:b8:c8	LiteonTe_e6:65:71	0x1234	222	Ethernet II	0,520648		Data Publishing - Frag2 Temperature		
350	267,9205196	Espressi_fc:b8:c8	LiteonTe_e6:65:71	0x1234	222	Ethernet II	0,401737		Data Publishing - Frag3 Temperature	0,922385282	
351	268,5343718	Espressi_fc:b8:c8	LiteonTe_e6:65:71	0x1234	222	Ethernet II	0,613852	1	Data Publishing - Frag1 Temperature		1,536237478
352	268,9417369	Espressi_fc:b8:c8	LiteonTe_e6:65:71	0x1234	222	Ethernet II	0,407365		Data Publishing - Frag2 Temperature		
353	269,4553462	Espressi_fc:b8:c8	LiteonTe_e6:65:71	0x1234	91	Ethernet II	0,513609		Data Publishing - Frag3 Temperature	0,920974378	
356	270,0680852	Espressi_fc:b8:c8	LiteonTe_e6:65:71	0x1234	222	Ethernet II	0,612739	2	Data Publishing - Frag1 Temperature		1,533713393
359	271,0959942	Espressi_fc:b8:c8	LiteonTe_e6:65:71	0x1234	222	Ethernet II	1,027909		Data Publishing - Frag2 Temperature		
	271,555414								Data Publishing - Frag3 Temperature	1,487328803	
360	272,0148338	Espressi_fc:b8:c8	LiteonTe_e6:65:71	0x1234	91	Ethernet II	0,91884	3	Data Publishing - Frag1 Temperature		1,946748607
361	272,5283978	Espressi_fc:b8:c8	LiteonTe_e6:65:71	0x1234	222	Ethernet II	0,513564		Data Publishing - Frag2 Temperature		
362	272,9357355	Espressi_fc:b8:c8	LiteonTe_e6:65:71	0x1234	222	Ethernet II	0,407338		Data Publishing - Frag3 Temperature	0,920901656	
365	273,5510909	Espressi_fc:b8:c8	LiteonTe_e6:65:71	0x1234	91	Ethernet II	0,615355	4	Data Publishing - Frag1 Temperature		1,53625712
366	274,0703594	Espressi_fc:b8:c8	LiteonTe_e6:65:71	0x1234	222	Ethernet II	0,519269		Data Publishing - Frag2 Temperature		
367	274,4746178	Espressi_fc:b8:c8	LiteonTe_e6:65:71	0x1234	222	Ethernet II	0,404258		Data Publishing - Frag3 Temperature	0,923526859	
372	276,1098556	Espressi_fc:b8:c8	LiteonTe_e6:65:71	0x1234	91	Ethernet II	1,635238	5	Data Publishing - Frag1 Temperature		2,558764653

Figure 5: Wireshark™ post-processed file in MS Excel™

.

lines, as shown in Figure 5. Column (A) contains a sequence number; column (B) has the *Timestamp* generated by the Wireshark™ application; columns (C) and (D) indicate the origin and destination of the collected packets, respectively. Columns (E) and (F) show the "*Ethertype*=0x1234" and the packet size, respectively; column (G) shows the information on the interface used. These fields help to obtain other columns (H. I, J, K and L), that have been manually post-processed. It is possible to observe in line 27, which is highlighted in yellow, that the third fragment of the second temperature measurement publication was lost. This figure shows how the statistics obtained from the tests were processed.

In addition to these two files ("data.txt" and "wireshark.csv"), three other log files fulfill the required analysis data: (i) Log_EPGS.txt from EPGS; (ii) Log_GW.txt from LoRa/Wi-Fi Gateway (GW); (iii) IoTTestApp_Core_Subscribe-RTT.dat from IoTTestApp. These logs were collected from the software installed on the laptop computer, facilitating the correlation between the spreadsheets, even though small variations in the timestamps have been detected. However, they do not represent a problem in the final analysis performed. These variations are due to the transmission or processing time of each phase or segment of the test performed. In addition, by evaluating the sequence numbering field (*count*) contained in the "data.txt" file, which is registered by

Table 3: Software and tools used in the experimental protocol.

Software	Function	Device	Data Format	Additional Notes
EPGS Software (embedded)	Collect sensor data and transmit NG message fragments over LoRa.	ESP-32	Logs in `Log_EPGS.txt`	Embedded in device firmware; provides sensor data for the generation of NG messages.
PGCS	Digital twin; NG message encapsulation on Wi-Fi and Ethernet; software router of NG messages.	Laptop	Logs are not used in the scenarios.	Represents ESP-32 and EPGS at the service level.
NRNCS	Name resolution service and network caching; divided in three services: PSS, GIRS, and HTS.	Laptop	Logs are not used in the scenarios.	Temporarily stores EPGS measured temperature samples. Notifies IoTTestApp when a new sample is available in the cache.
IoTTestApp	Receives NG IoT measurement data; stores statistics in a Linux local file	Laptop	`data.txt` (timestamp, node name, sequence number, sampled temperature)	Supports measurement correlation via sequence numbers. Final client of established contracts. Subscribes to temperature samples from NRNCS.
Wireshark™	Captures and filters NG network packets on Ethernet interface	Laptop	Captured packets, filtered by Ether-type=0x1234; exported as `wireshark.csv`	Filter minimizes dataset; supports detailed packet analysis and loss detection.
MS Excel™	Post-processes Wireshark logs and correlates data files	Laptop	`wireshark.csv` and others	Enables manual post-processing, filtering, and correlation of logs and packet data.
LoRa/Wi-Fi Gateway Firmware	Manages message translation and forwarding between LoRa and Wi-Fi.	ESP-32	Logs in `Log_GW.txt`	Supports monitoring and logging of message forwarding.
Teraterm	Open-source terminal emulator to operate LoRa node and gateway.	Laptop	Bootstraping, initialization, logs collecting, etc.	Supports operation and logging of firmwares.

the "IoTTestApp" application, it is possible to identify the NG messages lost in the path between EPGS and the application, during the data publication process to the NRNCS via PGCS (digital twin).

From all the information collected, it is possible to identify NovaGenesis message fragments that were unable to traverse all segments of the network, including LoRa, Wi-Fi and Ethernet. Then, we can calculate the loss of NG messages at any point since every NovaGenesis message is fragmented into three LoRa packets due to the 222 bytes limit. At the LoRa/Wi-Fi gateway (GW), the

NG fragments are housed in Wi-Fi frames, which are translated into Ethernet frames at the Wi-Fi access point near the laptop.

5. Performance Evaluation

This section presents our experimental results for NovaGenesis for LoRa including the following performance metrics: (i) message loss rate, (ii) out-of-order messages, and (iii) round-trip time (RTT) for IoT measurement subscription from the temporary network cache NRNCS. The test workload consists of samples from an existing temperature sensor on the ESP-32 of the LoRa node. All of these measurements were obtained according to the methodology and scenario in the previous section. Hence, all experimental data in this section were derived from a unified, measured performance evaluation scenario shown in Figures 2 and 3. Table 4 summarizes the experimental performance evaluations done in this section, as well as the practical use case that will be covered in Section 6.

Table 4: Summary of experimental performance evaluations and practical use case.

Test case	**Trials**	**Objective**	**Setup**	**Results**
Test 1	3	To determine NovaGenesis message and fragment loss performance in all network segments and end points.	Figure 2	Acceptable performance in terms of NovaGenesis message and fragment loss ratios.
Test 2	1	To perform radio frequency-based network analysis.	Figure 2	Level of RF interference level at the experimental site to assist in the other analyzes.
Test 3	1	To analyze out-of-order NovaGenesis messages.	Figure 2	To identify out-of-order messages and their impacts in communication.
Test 4	1	To determine round-trip IoTTestApp measure subscription delays from local network caching (NRNCS).	Figure 2	Highlight acceptable measure subscription delays from the network cache.
Use Case 1	1	Use case demo of the entire solution applied for environmental monitoring.	Figure 12	Demonstrate the feasibility of the solution for deployment in a practical setup within a commercial product.

5.1. Test 1 - NovaGenesis Message and Fragment Loss

This subsection presents the NovaGenesis message and fragment loss ratios, which are fundamental metrics in any low-power, wide-area, lossy network. It is worth pointing out that messages are made up of one or more fragments. Losing just one fragment leads to the total loss of the entire message. Table 5 shows the results of three experiments that we performed separately, that is, Experiments 1, 2, and 3. The number of hours required to perform each experiment (E) is reproduced in the second column. In the following column, we have the total number of NG messages transmitted in the experiment (Tx) and the number of messages successfully received (Rx).

Trial ID	Trial Duration (h)	Experimental data				Losses per data sampling point							
		Tx	Rx	Lost		EPGS		Node ⇔ GW		GW ⇔ Wi-Fi AP		IoTTestApp	
				Qty.	%	Qty.	%	Qty.	%	Qty.	%	Qty.	%
1	3.00	5402	5171	231	4.28	0	0	78	33.77	152	65.80	1	0.43
2	2.11	3806	3707	99	2.60	0	0	42	42.42	56	56.57	1	1.01
3	1.72	3103	3086	17	0.55	0	0	1	5.88	15	88.24	1	5.88
Total		12311	11964	347	2.82	0	0	121	34.87	223	64.27	3	0.86
Mean		6870	6761	110	1.60	0	0	37	34.09	71	65	1	0.91

Table 5: NovaGenesis message loss results measured at four places: EPGS, Node ⇔ LoRa/Wi-Fi Gateway (GW) link, GW ⇔ Wi-Fi AP link and IoTTestApp. Tx is the total number of NovaGenesis messages transmitted, while Rx is the number of successful NG received messages.

In Table 5 there is also the number of lost message fragments (Qty.) and its equivalent percentage (%). The last column contains detailed statistics on message losses per segment. The number of lost NG messages (Qty) and its equivalent percentage (%) is given for EPGS, Node ⇔ LoRa/Wi-Fi Gateway (GW) link, GW ⇔ Wi-Fi AP link, and IoTTestApp.

The worst-case fragment loss occurred in Experiment 1, with 231 fragments lost, resulting in 231 NG messages lost. This is equivalent to a message loss rate of 4.28%. The other two experiments had lower message loss rates, that is, 2.60% and 0.55%, respectively. If we consider all trials, we have a loss of 2.82% in total. If we calculate the average message loss rate, we have 1.60%. When we look at the different points on the network, we can see that the highest mean message loss rate (65%) was detected by the Wireshark^{TM} analysis in the Ethernet segment. This means that a predominant amount of fragments were lost when the Wi-Fi link was crossed. The mean message loss rate in the LoRa segment was quite inferior, that is, 34.09%. About 0.91% was lost on the path that starts at the Wi-Fi access point and ends at the IoTTestApp application, that is, the Ethernet segment. No message was lost in EPGS on the ESP-32 IoT node. Based on these results, it can be stated that the Wi-Fi link caused losses 191.89% greater than the LoRa link.

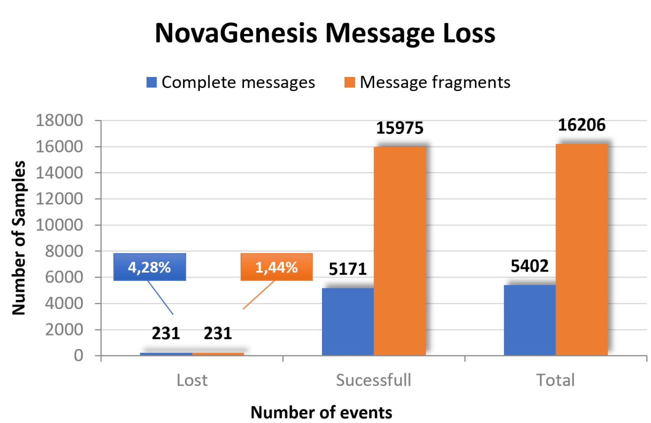

Figure 6: Received and lost NG messages and their fragments. Messages are made up of one or more fragments. Losing just one fragment leads to the total loss of the entire message.

Figure 6 shows the results for trial 1 that include the total number of fragments of the NG message. A loss of 1.44% of the fragments of the NG message leads to a loss of 4.28% of the messages. Each NG message is known to be transmitted in three fragments because of the LoRa characteristic, and it was observed in the collection analysis that the loss of an NG message is associated with the loss of only a single segment. The total number of message fragments in this trial was 16206 during 3 hours, with EPGS sending temperature measures every 2 seconds.

Figure 7 presents the results obtained from the analysis of the collected *logs* for the same Trial 1. Despite the complexity of implementing NG in a LoRa network, with a Wi-Fi access point and an Ethernet network segment, NovaGenesis (EPGS and Core) did not affect the loss rates in this proof-of-concept scenario, being responsible for **0.00%** and **0.43%**, respectively. The LoRa segment, between EPGS and GW, represented **33.77%** of total losses. That is, of the 231 identified losses, 78 are related to the LoRa segment. The losses in the Wi-Fi segment were 152 (65.68%).

Table 6 presents a more detailed view of fragment losses for experimental trials. Losses do not affect NovaGenesis message fragments equally. We mentioned earlier that all lost messages had three fragments. We know that the more fragments a message has, the greater the probability of loss. As there is no retransmission, it is enough to lose a fragment to lose the entire message. The greatest proportion of fragments lost are the second, accounting for 36.31%, followed by the third at 34.87%, and the first at 28.81%. We did not find a specific reason for these statistics. To find a possible cause, it is necessary to investigate the Wi-Fi and LoRa links and the gateway itself. However, this is a subject for future work. Table 6 also shows how many fragments were lost consecutively in each trial. A total of 3.17% of the fragments suffered consecutive losses.

5.2. Test 2 - Radio Frequency Analysis of the Experimental Location

An additional test was performed to identify Wi-Fi usage in the location of the experimental trials. An Aruba AP model IAP-205RW [34], with *firmware*

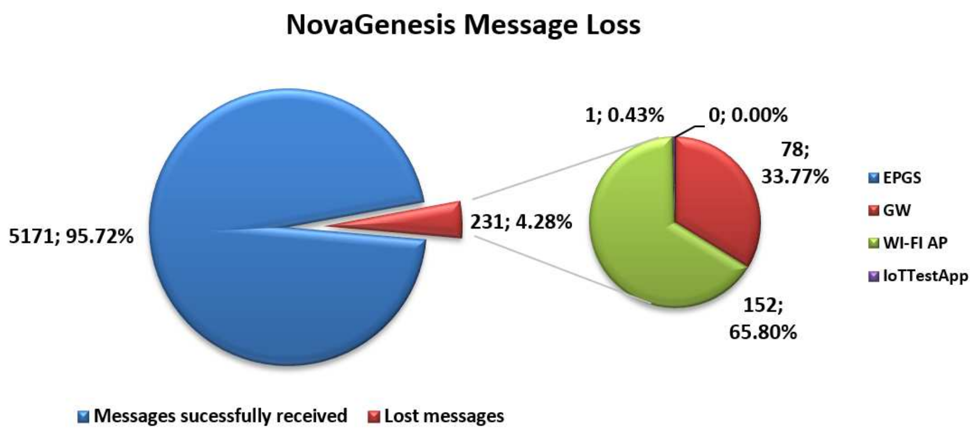

Figure 7: NG messages loss in an NG over LoRa/Wi-Fi/Ethernet network.

Trial ID			Fragment loss							
	Message loss		Fragment 1		Fragment 2		Fragment 3		Consecutive loss	
	Qty	%	Qty	%	Qty	%	Qty	%	Qty	%
1	231	4.28	71	30.74	82	35.5	78	33.77	10	4.33
2	99	2.6	24	24.24	33	33.33	42	42.42	1	1.01
3	17	0.55	5	29.41	11	64.71	1	5.88	0	0
Total	347	2.82	100	28.81	126	36.31	121	34.87	11	3.17

Table 6: Detail of message and fragment loss results for the performed trials, including consecutive message losses. The disappearance of even a single fragment results in the complete loss of the entire message.

version equal to 6.5.4.22 and configured in *Spectrum Monitor* mode, offered spectrum analyzer functionality for non-licensed bands. It was possible to assess the level of interference in Wi-Fi for the 2.4 and 5.8 GHz frequencies on Paulista Avenue in São Paulo. The result of this spectrum scan provided relevant information for this article. Figure 8 reproduces excerpts from the AP management web page. We can observe that:

(1) This screen displays all radiating devices identified within the AP's coverage radius, which operate at the 2.4 GHz frequency, and which may be causing RF interference to our network.

(2) This graph illustrates the quality of the channels and their noise level for the 2.4 GHz frequency band. It is possible to observe that more than

Figure 8: Level of RF interference at the experimental testing site. The radiating devices are identified in (1). In (2), the quality of the channels and their noise level are presented. The amount of neighboring APs and clients is shown in (3). Channel utilization at 5.8GHz is shown in (4).

40% of channel 11 suffers from noise and interference problems, which can cause packet loss.

(3) *Access Point* is able to quantify the devices identified within the coverage radius divided into two groups: one that lists the APs that cause interference in the neighborhood and another that indicates the number of clients that also generate interference on the network, represented by the amounts of 223 APs and 53 clients, respectively.

(4) And finally, analyzing the graph generated by the AP at the 5.8 GHz frequency, it is possible to identify a much lower level of interference.

Based on the analysis of these measurements, it can be stated that the significant number of devices radiating at the 2.4 GHz frequency directly affected packet loss during our experiment. Despite such high packet losses, NG self-organization procedures were properly completed, allowing EPGS to expose its functionalities, receive a service offer from IoTTestApp, and accept the offer to publish periodic temperature samples to the NG application. Such service-oriented, programmable, and contract-based operation of IoT devices and services makes sensing solutions more flexible, reducing human intervention, and exposing IoT devices' capabilities dynamically via natural language. In conclusion, named IoT services and content can be used with resource-limited technology such as LoRa.

5.3. Test 3 - NovaGenesis Out-of-Order Messages

By examining the individual spreadsheets from the single file created after processing the EPGS records/*logs*, LoRa/Wi-Fi Gateway, Wireshark™ statistics in the Ethernet segment after Wi-Fi AP, and "data.txt" from NG IoTTesTApp, it was possible to determine if there were any reversals in the sequence of the NG messages and/or their fragments. For Trial 1, only 31 of the 5402 data publishing messages had at least one fragment of an NG message that reached the NG Core out of sequence, which is equivalent to **0.57%** of the total samples, as shown in Figure 9. A thorough examination of the Wireshark™ *logs* allowed us to identify out-of-order problems. We were able to do this by looking at the packets that were sent over the Ethernet network.

Delays in NG message fragments cause data publications to occur out of order. By processing the "data.txt" file at the final destination (IoTTestApp), it was possible to identify such out-of-order problems since for each record, the sequence number counter (*count*) sent by the EPGS is saved and compared. PGCS has the ability to reassemble fragments out of order. IoTTestApp is capable of reorganizing the order of received temperature measurements at EPGS. Both features demonstrate NG's capability to handle out-of-order fragments and messages.

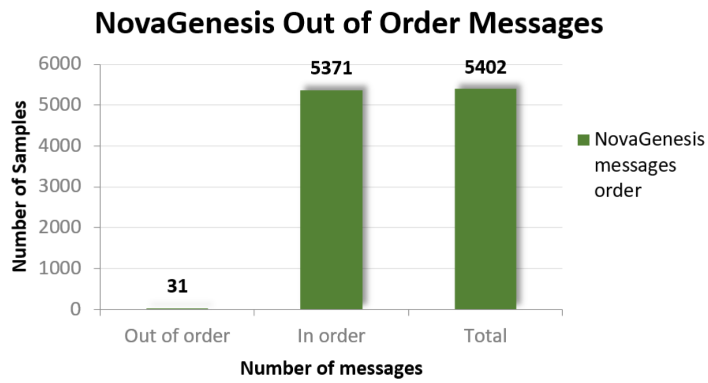

Figure 9: NovaGenesis out of order messages.

NG and EPGS had no difficulty adjusting messages to other communication channels, such as the LoRa network, which limits the size of the PDU to 222 bytes. In our previous experiment using Wi-Fi [2], the size of the PDU could accommodate the entire NG message, that is, 1200 bytes. The NG adaptation layer allows for the use of Maximum Transfer Units (MTUs) of a few bytes up to 33 Mbytes (adjustable).

It was not possible to identify the factors that led to the inversion of the fragments. However, these were not significant enough to impact the data publication service, since after receiving the fragments and reassembling the packets/messages by the NG, the data are delivered to the subscriber of the service, even if delayed. Investigating this issue could be a subject of study for future work.

5.4. Test 4 - Round-Trip Time for IoT Measurement Subscription

The performance of the service provided by NG is measured by the difference in the delivery times of a new data publication, which is received by the PSS and then notified to the consumer application "IoTTestApp". The data stored in the file "IoTTestApp_core_SubRTT.dat" is depicted in Figure 10. The analysis of the data in this file revealed that the average RTT for IoTTestApp when subscribing to a temperature sample in PSS/GIRS/HTS was approximately **38.02**

milliseconds. The graph shows a blue line with asterisks labeled "Trend" that indicates the mean value.

During the test period, NovaGenesis software, Wireshark^{TM}, and Teraterm all run at the same time. The notebook processing rate was observed to reach 100% once the NG was up and running and remained that way for the duration of the test. This can explain the RTT values in Figure 10.

One hundred samples from a total of 5042 were chosen to gain an understanding of the experiment's performance. Each sample was taken at equal intervals from the entire set of collected measurements and represented a row from the processed spreadsheet of the file "IoTTestApp_Core_SubRTT.dat". The instantaneous results of the consolidated samples are shown in the graph in Figure 10. The selection of 100 samples was made with the intention of providing a visual representation of the indicators and data that can be obtained from this document without compromising its essence or content. Having a large number of samples would make it impossible to visualize.

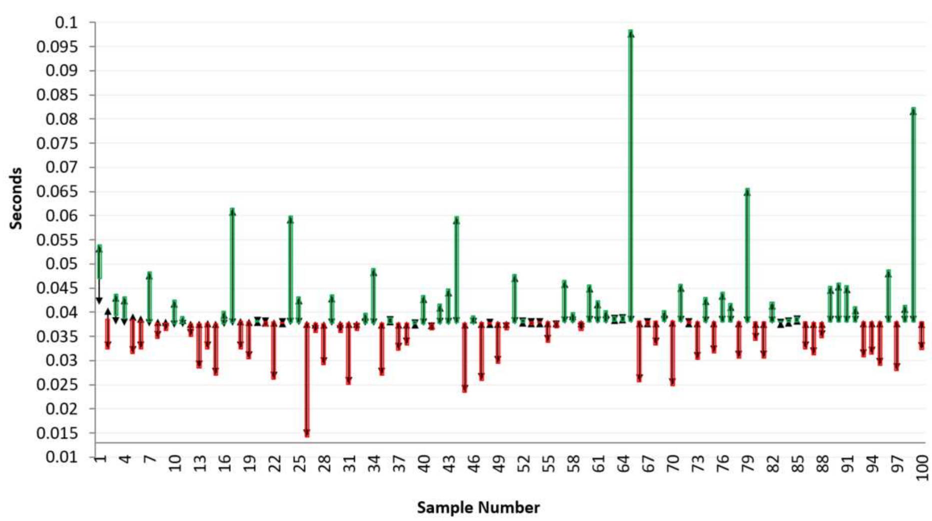

Figure 10: Instantaneous RTT measured at NG's IoTTestApp Application when subscribing a temperature sample from the PSS in the laptop.

When studying the highest points shown in Figure 10, such as samples 16, 23, 26, 44, 65, etc., they suggest that they are preceded by a message loss incident.

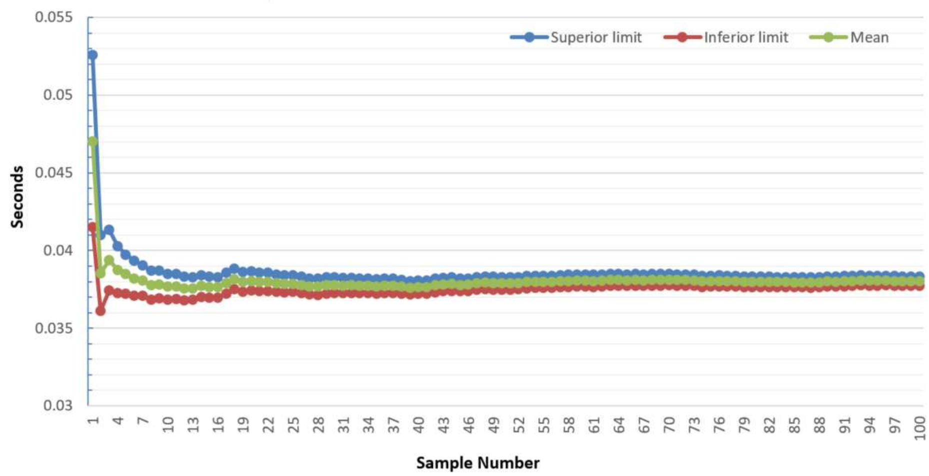

Figure 11: Mean RTT measured at NG's IoTTestApp Application when subscribing a temperature sample from the PSS in the laptop.

It is essential to have a trend indicator with average values, standard error, and confidence intervals in order to assess the quality of the service provided. The results of NovaGenesis and its applications show that they are capable of providing contracted services with pertinent and reliable data, which is an integral part of their design.

Figure 11 illustrates the average (mean) RTT for subscribing to a measured room temperature sample, derived from each chosen instantaneous measurement throughout the data set. As evidenced by the data, the mean RTT is approximately 38.02 ms, which is deemed an acceptable value in the context of this application.

6. Use Case 1: Environmental Data Monitoring

Figure 12 depicts an actual scenario designed to collect environmental data utilizing EPGS in Santa Rita do Sapucai, Minas Gerais, Brazil. The meteorological station used in the experimental scenario is depicted in Figure 13. The original control module provided by the company Ativa Soluções was replaced with an ESP-32 module (LoRa node) that incorporates EPGS, as illustrated in Figure 14. The weather station established communication with the LoRa/Wi-Fi gateway housed in another ESP-32. This setup, in turn, facilitates communication with the laptop on which the NovaGenesis PGCS, NRNCS, and IoTTestApp operate via Wi-Fi. The following sensors were used in the experiment that lasted approximately 24 h: (i) Rain gauge (rain); (ii) TUP (temperature, humidity, and pressure); (iii) Anemometer (wind speed/direction); (iv) Pyranometer (solar radiation); and (v) UV (ultraviolet radiation). In addition to these environmental metrics, the RTT expended by the IoTTestApp to subscribe to an environmental sample from the local HTS (part of NRNCS) was also measured.

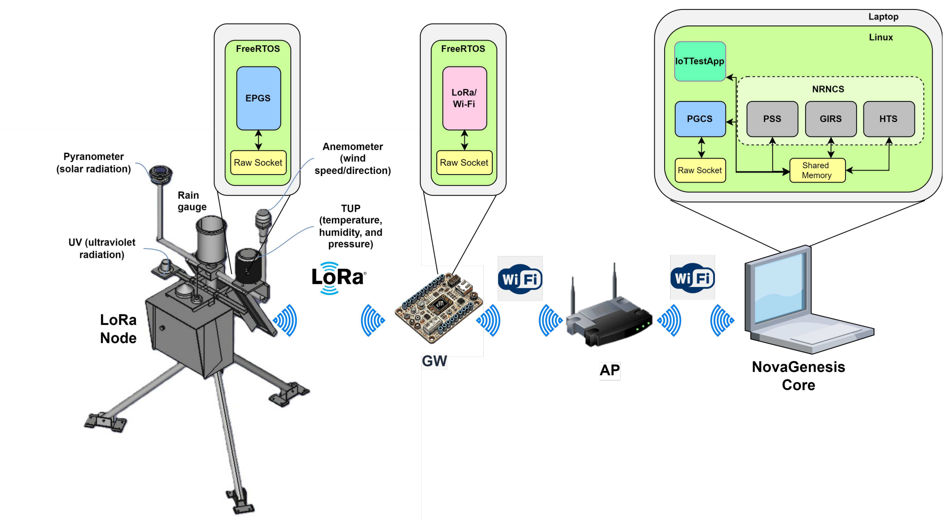

Figure 12: Ilustration of the experimental scenario for environmental monitoring.

6.1. Evaluation Methodology

The experiment was initiated at 4 p.m. on October 8th, 2024, and persisted for an estimated duration of 24 hours. Throughout the duration of the experiment, no precipitation was recorded, which consequently resulted in the rain gauge registering the absence of measurable precipitation. The calculation of the mean utilizes the cumulative samples obtained, beginning with the initial

Figure 13: Image depicting the meteorological station used in the experimental setup at the Escola Técnica de Eletrônica (ETE) Francisco Moreira da Costa, Brazil.

sample and progressing to the current sample along the x-axis. The results include a confidence interval to modify the error bars, employing the constant *1.96* to establish the error margins relative to the mean value. To reduce the dataset size and improve the clarity of the visualization, we performed subsampling by selecting one data point every 50 samples from the original dataset.

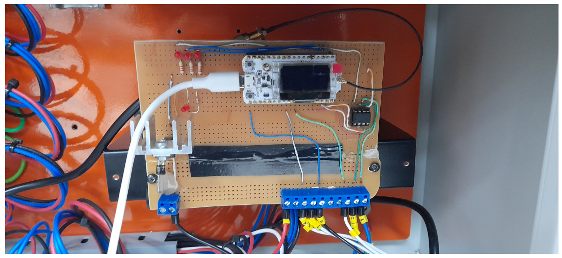

Figure 14: Developed control module incorporating NovaGenesis. This module operates with the ESP-32 utilizing the EPGS for LoRa communication and goes inside the box in the previous figure.

6.2. Results

Figure 15 presents the temperature dynamics from October 8th-9th, 2024, illustrating the typical diurnal cycle of warming and cooling. Temperatures start high (33 to 36°C) in the late afternoon, followed by a sharp decline after sunset. Overnight, the temperature drops to 18–19°C, remaining stable for a few hours before steadily rising with the onset of the day. This pattern reflects the tropical/subtropical climate of Santa Rita do Sapucaí (SRS), Minas Gerais, Brazil. The graph incorporates individual temperature samples (red crosses) and mean values (blue curve) with small error margins, indicating precise sensor readings.

Figure 16 shows the diurnal cycle of relative humidity. It rises overnight to 100% in the morning and decreases as the day warms. Starting around 40%, this pattern reflects how cooler air retains less moisture, typical of SRS, where nights are more humid. The graph features individual samples (light blue crosses) and mean values (green curve) with small error bars, demonstrating accurate sensor performance.

Figure 17 illustrates atmospheric pressure changes over 24 hours. Starting at 918 hPa, the pressure increases as the temperatures drop at night and peaks at 922 hPa during the day, reflecting typical diurnal patterns. The graph includes discrete pressure points such as orange crosses, a purple mean curve, and error bars, confirming the stability and reliability of the sensor.

Figure 18 shows 24-hour wind speed measurements, presenting a steady pat-

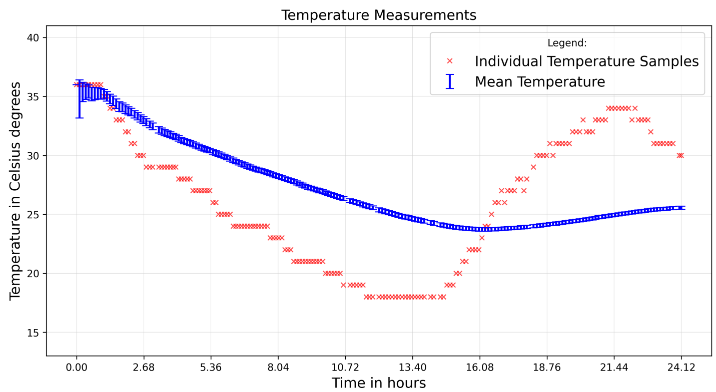

Figure 15: Mean temperature over 24h.

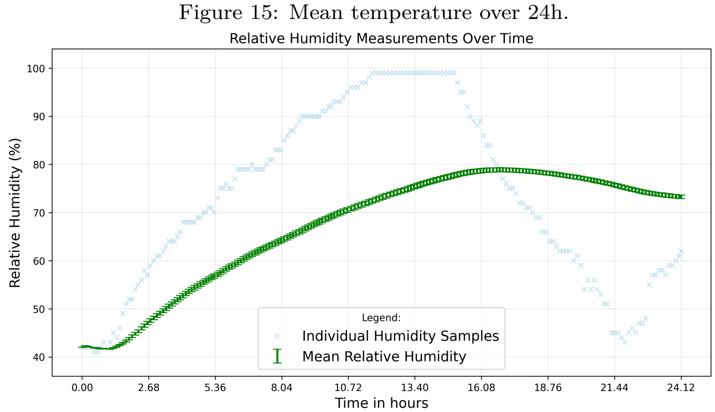

Figure 16: Mean relative humidity over 24h.

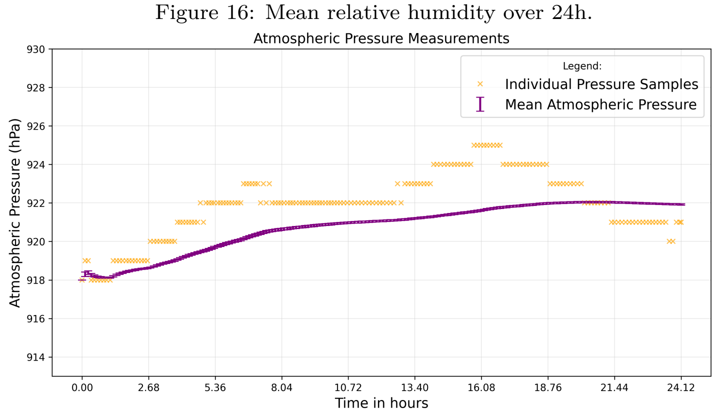

Figure 17: Mean atmospheric pressure over 24h.

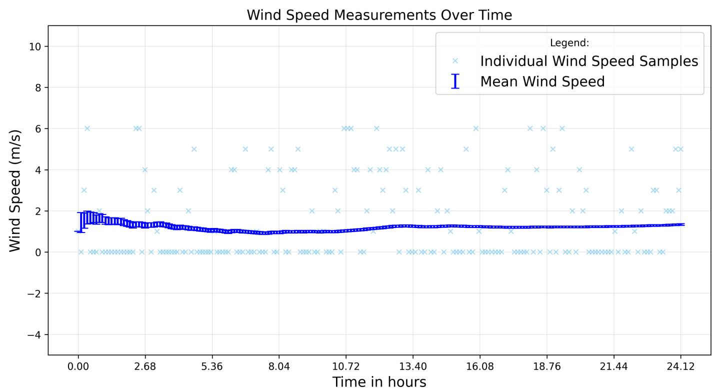

Figure 18: Mean wind intensity over 24h.

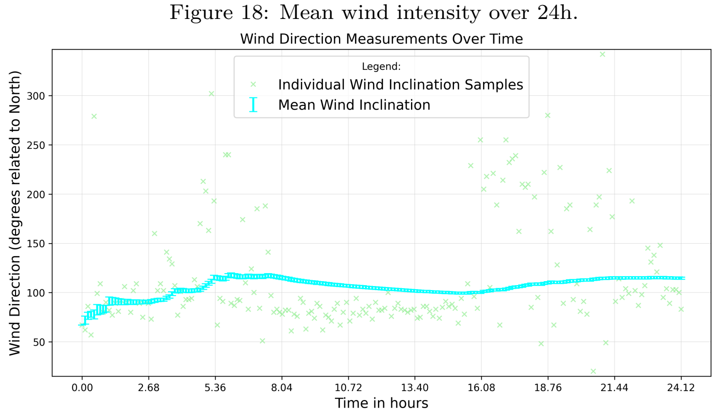

Figure 19: Mean wind inclination over 24h.

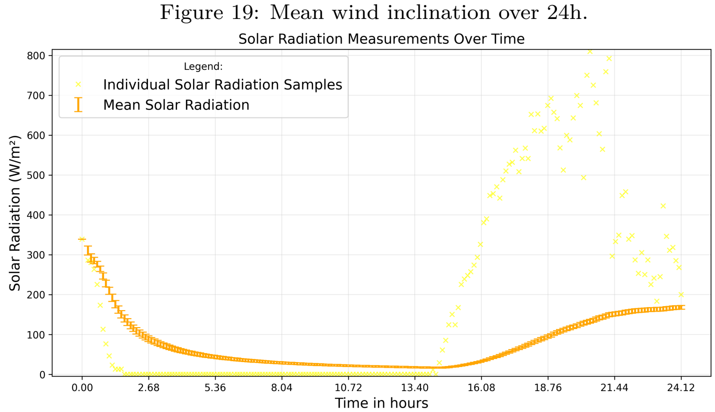

Figure 20: Mean solar radiation over 24h.

tern. Speeds average around 1.5 m/s with occasional gusts up to 6 m/s, typical for calm conditions recorded by the station. Individual speeds are light-blue crosses, with the mean speed as a dark-blue curve with small error bars.

Figure 19 displays the wind direction relative to the north over 24 hours. Individual values (light green crosses) vary greatly, with angles from nearly 0° to 350°, showing wind direction changes. The mean inclination (cyan curve) is stable, between 100° and 150°. Small error bars suggest reliable sensors, which capture quick shifts and stable patterns, consistent with expected local wind behaviors in SRS.

Figure 20 illustrates solar radiation over 24 hours, showing a typical diurnal cycle. Radiation peaks near 700 W/m^2 in the afternoon and drops to almost zero at night and early in the morning. There is a sharp rise at sunrise and a gradual decrease at sunset. The orientation of the sensor affects the readings. Daily samples (yellow crosses) vary, but mean values (orange curve) with small error bars confirm sensor reliability and stability.

Figure 21 displays a 24-hour UV index pattern in SRS, starting at zero at night, rising rapidly post-sunrise, peaking at over 300 in the afternoon, and then declining towards sunset. This behavior is typical in clear skies. Individual samples (purple crosses) vary more at peak hours, while the mean values (pink curve) with small error bars show consistent sensor performance.

Figure 22 illustrates both individual and average RTT samples for the transmission of a measurement file from HTS to IoTTestApp, as executed on the laptop depicted in Figure 12. The average RTT was calculated to be approximately 36.74319 ms, with a margin of error of 0.18493 ms. This value is deemed suitable for most IoT applications, including environmental monitoring.

In summary, the aforemetioned results reveal different diurnal patterns: the temperature fluctuated between 18°C and 36°C, humidity peaked near 100% during the night, and solar radiation and the UV index showed typical surges during daylight hours. Wind speed remained generally low with occasional gusts, while wind inclination demonstrated notable variability. The atmospheric pressure increased slightly overnight, peaking at 922 hPa. Individual sample points, along with their arithmetic means and small error margins, confirm the accuracy and stability of the sensors in all measurements.

These results provide valuable information on local weather behavior and demonstrate the reliability of the station's data collection setup and communication architecture. They highlight the adequacy of the NovaGenesis platform over LoRa/Wi-Fi for a practical environmental monitoring use case implemented in a commercial Ativa Soluções Ltd product. The solution could transport multiple sensor measurements for approximately 24 hours with acceptable delays. The steps presented in Section 3.2 for the self-organization of named services and the publishing/subscribing of named content have been successfully demonstrated.

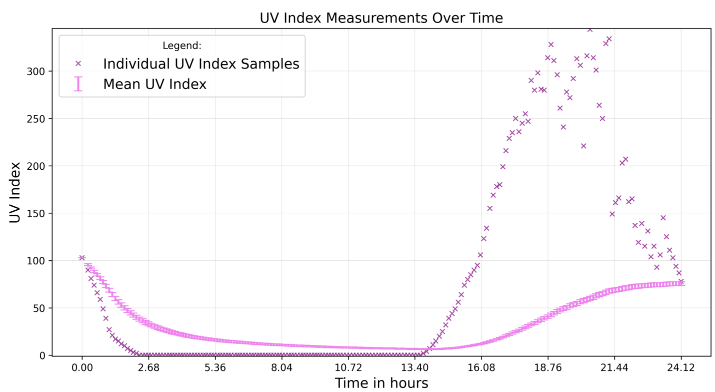

Figure 21: Mean UV index over 24h.

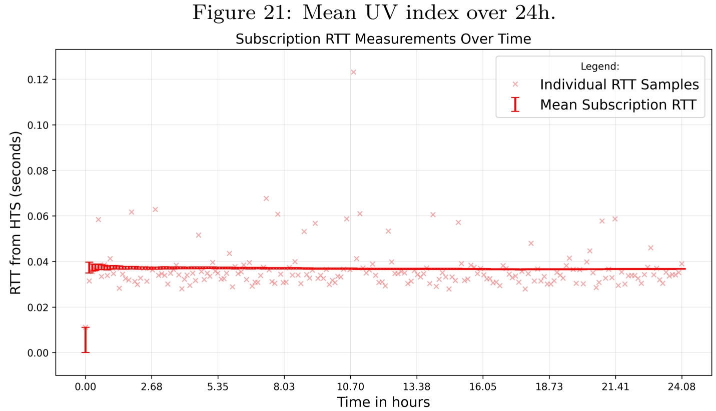

Figure 22: Mean subscription RTT from HTS to IoTTestApp over 24h.

7. Discussion, Lessons Learned, and Future Work

Table 7 summarizes the main challenges faced in this work, the solutions implemented and the future work planned. The challenges started with the LoRa itself. After some initial analyzes that took into account the demands of NovaGenesis and the testing scenario, the ESP-32 node (where EPGS runs) was configured to operate in a LoRa Point-to-Point communication mode. Since the Espressif IoT Development Framework (ESP-IDF) is based on FreeRTOS, another challenge was to implement task scheduling and efficient resource management. The main application implemented a continuous listening loop through a dedicated FreeRTOS task that repeatedly set the radio transceiver to receive mode, enabling packet detection. For data transmission, the system explicitly invoked a function to send packets, momentarily switching the transceiver from receive to transmit mode and back. This configuration provided low-level control over radio operation, offering flexibility for the NG implementation.

Another challenge was to deal with losses in the LoRa and Wi-Fi network segments. The LoRa reception logic included checks for Cyclic Redundancy Check (CRC) errors to discard corrupted packets. The reception task ran continuously and unblockingly, allowing for the timely processing of valid packets. Fragmentation and reassembling was implemented in EPGS and PGCS. To reduce the impacts of fragment loss in these frequency bands, an option would be to use multiple message transmission techniques, as implemented in some scenarios in CAT-M and NB-IoT [35]. In the current EPGS code protocol, the retransmission of lost NG messages is not anticipated as a result of the need to preserve battery life. Implementing multiple transmissions would increase the probability that a message would reach its destination, but it would reduce the energy efficiency of the IoT node. More work is required to evaluate a retransmission function in the NG and EPGS, which is not supported in the current code.

Another option is to address lost messages at the application layer (IoTTest-App) by calculating an average value, since temperature measurements are stable values and do not fluctuate suddenly. One point that reinforces this suggestion of average calculation is that the number of consecutive losses of the NG message (two simultaneous measurements) was relatively low, as shown in Table 6. An alternative strategy to improve performance involves optimizing the transmission parameters. Consequently, NG timer optimization can be used to more effectively leverage the capabilities of the LoRa network.

It should be noted that the tests were conducted with a data publication cadence (temperature measurements) every 2 seconds. For cases where the transmission frequency is spaced in which data samples would be sent every hour or even longer intervals, the implementation of multiple sequential publications can help reduce or even eliminate the possibility of not having the data at the desired time. The number of multiple transmissions can vary depending on the scenario. The identification of duplicate messages or the use of the one containing the complete message can be a reason for implementation in the NG itself, similar to the types of network classes supported in LoRa. Moreover, the evaluated scenario assumes a constant LoRa link length. The tests carried out were confined to short distances, focusing on proof of concept under controlled conditions. Future investigations would explore the impact of varying the distance between the nodes and the gateway. Another issue is handling the mobility of LoRa devices.

A constraint of the existing implementation is the concurrent utilization of multiple LoRa devices within the same network infrastructure. Consequently, a Medium Access Control (MAC) paradigm is required to regulate access to the LoRa medium, while queueing and QoS policies manage contention on the Wi-Fi backhaul. An immediate enhancement could be achieved through simultaneous multi-message reception at the LoRa/Wi-Fi gateway by employing multi-channel, multi-SF concentrators; however, this would require hardware beyond the current ESP32-based gateway. On individual LoRa nodes equipped with a single transceiver, concurrency must still be handled at the software level through interrupts, buffering, and task scheduling rather than true parallel RF demodulation. Implementing this on IoT devices such as the ESP32 is particu-

Table 7: Challenges and solutions in NovaGenesis over LoRa/Wi-Fi communication.

Current Work	
Challenge Description	**Approach Implemented**
Maintaining continuous reception without blocking other tasks.	A dedicated FreeRTOS task (task_rx) handles LoRa reception. It continuously checks for received packets using lora_rxed() and processes them. Crucially, the vTaskDelay(1) call at the end of the loop ensures that the task yields control to the FreeRTOS scheduler, allowing other tasks to run and preventing it from blocking the system. This effectively achieves continuous reception without blocking other operations.
Balancing transmission and reception states efficiently.	The transceiver is explicitly switched between RX and TX modes as needed.
Limited LoRa payload size of 222 bytes.	Implemented transparent fragmentation and reassembly by EPGS and PGCS.
Leverage the dual-core processing capabilities of the ESP32	Implemented distinct tasks for LoRa transmission (task_tx) and reception (task_rx), pinning each to a separate ESP32 core (Core 0 and Core 1, respectively) to achieve parallel processing and efficient multitasking.
Lack of support for simultaneous multi-message reception.	PGCS was improved to deal with parallel reception buffers and multithread socket reading.
Self-verifying identifiers for LoRa devices, RTOS, and EPGS (E1 and E5).	EPGS implements the NovaGenesis protocol defining message structures (in components like ng_message.c and ng_command.c) that incorporate unique identifiers. Hashing utilities (indicated by ng_epgs_hash.c) are employed to generate and verify unique identifiers, ensuring data integrity and authenticity. Data serialization using JSON (ng_json.c), is used to package these identifiers within messages. Received messages are processed and validated, likely including identifier checks, within functions such as newReceivedMessage.
IoT named data network caching and ID/Loc resolution (E2).	We equipped EPGS with the capability to publish/subscribe named content to/from the NRNCS.
LoRa device representation at service level (E3).	Implemented digital twin function on PGCS.
Contract-based operation and services self-organization (E4 and E5).	Expanded PGCS, IoTTestApp, and EPGS to deal with LoRa contracts.
Future Work	
Challenge Description	**Possible Approach**
Consecutive loss of NG messages (two simultaneous measurements).	Calculate average values at application layer due to stable temperature readings.
NG message fragment loss	Improve retransmission mechanisms or use message duplication detection within NG.
No retransmission mechanism in current EPGS code to preserve battery life.	Implement multiple transmissions; trade-off between energy efficiency and reliability.
Handling of sporadic message transmissions.	Use sequential multi-transmissions to improve batery life and reduce data loss probability.
LoRa node mobility.	Use periodic presence re-advertisement and automatic routing update mechanisms.
Concurrent use of multiple LoRa devices on the same network.	Introduce medium access control (MAC).

larly challenging due to the need for efficient concurrency management. While the ESP32 supports FreeRTOS-based multitasking and a dual-core architecture, developers must carefully manage memory, interrupt priorities, and timing constraints to avoid packet loss and ensure real-time responsiveness. Additionally, to support LoRaWAN operation on the ESP32, the current implementation would need to be extended with a compatible LoRaWAN stack, such as an ESP-IDF-adapted version of LoRaMAC-Node.

In addition, gateway and network-level load-balancing strategies are required to efficiently manage multi-device concurrency and maintain network stability in dynamic IoT topologies. By distributing traffic across multiple gateways and selecting optimal downlink paths based on received signal strength, signal-to-noise ratio, latency, and gateway availability, these mechanisms would prevent bottlenecks and optimize resource utilization, particularly in scenarios with high-density LoRa nodes. This functionality could be implemented as a new NovaGenesis service, enabling software-defined orchestration. However, the implementation of a control plane would be necessary on the network devices. This is something that has already been done in previous works with NovaGenesis. Furthermore, adaptive load-balancing mechanisms can significantly improve mobile node handover performance by dynamically reallocating traffic according to network conditions. Integrating these strategies with Wi-Fi Quality of Service (QoS) mechanisms is essential to prioritise critical IoT traffic, ensuring low latency and reliable throughput even under varying conditions. Finally, unique self-verifying IoT device IDs embedded in NG headers can be leveraged to differentiate traffic, enforce policies, and better handle device mobility.

Compared to related work, our NovaGenesis-based LoRa IoT approach advances state-of-the-art by provinding devices representation via digital twin (E3) and IoT service orchestration with contract-based operation (E4). In contrast, these characteristics are conspicuously absent in the extant studies reviewed. However, our proposal also covers other domains covered by alternative Internet proposals, such as NDN, XIA, and RINA. More specifically, our solution includes a general naming and name resolution approach integrated with network caching for IoT samples (E1), ID/Loc splitting for device mobility without compromising identity (E2), and the provenance and traceability of data samples (E5). The establishment of trust networks among devices and services constitutes an additional advantage inherent to the NovaGenesis paradigm.

8. Conclusion

This work proposed the expansion of NovaGenesis to support advanced IoT services using a long-range communication medium (LoRa). The difference in this study lies in NovaGenesis's ability to operate efficiently in IoT scenarios that utilize LPWAN, even when subjected to high RF interference. Our proof-of-concept integrated functionalities such as self-verifiable names, name bindings, semantic rich resource exposition, self-discovery, contract establishment, network caching, and services lifecycle, all essential for publishing data by EPGS over LoRa/Wi-Fi/Ethernet. Furthermore, the embedded version of NovaGenesis did not demonstrate significant performance degradation, validating its robustness and adaptability to the LPWAN domain.

Section 5 aligns with the hypothesis outlined in the Introduction and verifies the contributions identified as 1 and 2 within this section. We conclude that NovaGenesis performed without difficulties with limited PDU messages in LoRa, demonstrating the ability to adapt to different physical environments through packet fragmentation, reassembly, and reordering management. The "IoTTestApp" application leveraged the core features of NovaGenesis to measure the level of service delivered, presenting robust performance indicators that support new IoT services and business models. The PGCS functioned according to expectations to provide the digital twin capability for sensor devices. It was unexpected to discover that the majority of the packet loss occurred in the Wi-Fi segment of the network. In the environmental monitoring experiment conducted in Santa Rita do Sapucaí, Brazil, over 24 hours, the collected data showed high precision and stability, with metrics such as temperature ranging from 18°C to 36°C, humidity reaching nearly 100% overnight, and significant variations in the direction of the wind, confirming the reliability of the weather station and the effectiveness of the NovaGenesis platform over LoRa.

Despite the progress achieved, this study presents some limitations that pave the way for future work. The proposed architecture still needs to be tested with a larger number of nodes and in a long-distance (about 100 km) farm to assess its scalability. This scenario represents a compelling case study for the application of 5G and 6G technologies. Additionally, the implementation of message retransmission in the NG layer and the addition of a security layer are essential demands to be addressed. The complete development of the solution as a market-ready product and its standardization by bodies such as the IETF, IEEE, and ETSI are crucial steps for widespread adoption of the proposal. Future work should also consider forming a consortium of stakeholders to facilitate the formal recognition process and ensure that the solution meets market requirements.

Generative AI

During the preparation of this work the author(s) used ChatGPT and Writefull tools in order to verify English grammar. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

Acknowledgments

This work was partially supported by RNP, with resources from MCTIC, Grant No. 01245.020548/2021 − 07, under the Brazil 6G project of the Radiocommunication Reference Center (Centro de Referência em Radiocomunicações - CRR) of the National Institute of Telecommunications (Instituto Nacional de Telecomunicações, Inatel), Brazil. The work has also been partially funded by xGMobile – EMBRAPII-Inatel Competence Center on 5G and 6G Networks, with financial resources from the PPI IoT/Manufatura 4.0 from MCTI grant number 052/2023, signed with EMBRAPII. This work was carried out with support from CNPq, National Council for Scientific and Technological Development - Brazil (303382/2021-0). Furthermore, it is supported by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) - Finance Code 001. The authors also thank SINDISAT, Ativa Soluções, CNPq, FAPEMIG, and Vivavox Telecom.

References

[1] G. Mulligan, The 6LoWPAN architecture, ACM Journals (2007).

[2] A. M. Alberti, G. D. Scarpioni, V. J. Magalhaes, A. C. S. Jr., J. J. P. C. Rodrigues, R. da R. Righi, Advancing novagenesis architecture towards future internet of things, IEEE Internet of Things Journal 6 (2019) 215 – 229.

[3] T. B. da Silva, F. L. Verdi, J. C. G. de Melo, J. A. Suruagy, F. Silva, A. M. Alberti, Toward next-generation and service-defined networks: A novagenesis control agent for future internet exchange point, IEEE Network 36 (2022) 74–81.

[4] A. M. Alberti, M. M. Bontempo, J. R. Dos Santos, A. C. Sodré, R. D. R. Righi, Novagenesis applied to information-centric, service-defined, trustable iot/wsan control plane and spectrum management, Sensors 18 (2018).

[5] A. M. Alberti, G. D. Scarpioni, V. J. Magalhães, A. Cerqueira S., J. J. P. C. Rodrigues, R. da Rosa Righi, Advancing novagenesis architecture towards future internet of things, IEEE Internet of Things Journal 6 (2019) 215–229.

[6] A. M. Alberti, M. A. F. Casaroli, D. Singh, R. da Rosa Righi, Naming and name resolution in the future internet: Introducing the novagenesis approach, Future Generation Computer Systems 67 (2017) 163–179.

[7] A. M. Alberti, D. Mazzer, M. Bontempo, L. H. de Oliveira, R. da Rosa Righi, A. Cerqueira Sodré, Cognitive radio in the context of internet of things using a novel future internet architecture called novagenesis, Computers Electrical Engineering 57 (2017) 147–161.

[8] A. M. Alberti, M. A. F. Casaroli, D. Singh, R. da Rosa Righi, Naming and name resolution in the future internet: Introducing the novagenesis approach, Elsevier 67 (2016) 163–179.

[9] M. Adnan, I. Ahmed, S. Iqbal, M. R. Fazal, S. J. Siddiqi, M. Tariq, Exploring the convergence of metaverse, blockchain, artificial intelligence, and digital twin for pioneering the digitization in the envision smart grid 3.0, Computers and Electrical Engineering 120 (2024) 109709.

[10] D. Han, A. Anand, F. Dogar, B. Li, H. Lim, M. Machado, A. Mukundan, W. Wu, A. Akella, D. G. Andersen, J. W. Byers, S. Seshan, P. Steenkiste, Xia: Efficient support for evolvable internetworking, NSDI'12, USENIX Association (2012) 23–23.

[11] S. Vrijders, D. Staessens, D. Colle, F. Salvestrini, E. Grasa, M. Tarzan, L. Bergesio, Prototyping the recursive internet architecture: the irati project approach, IEEE fNetwork 28 (2014) 20–25.

[12] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, kc claffy, P. Crowley, C. Papadopoulos, L. Wang, B. Zhang, Named data networking, ACM Journals (2014) 66 – 73.

[13] J. de Carvalho Silva, J. J. P. C. Rodrigues, A. M. Alberti, P. Solic, A. L. L. Aquino, Lorawan — a low power wan protocol for internet of things: A review and opportunities, in: 2017 2nd International Multidisciplinary Conference on Computer and Energy Science (SpliTech), pp. 1–6.

[14] T. B. da Silva, R. P. dos Santos Chaib, C. S. Arismar, R. da Rosa Righi, A. M. Alberti, Toward future internet of things experimentation and evaluation, IEEE Internet of Things Journal 9 (2022) 8469–8484.

[15] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, k. claffy, P. Crowley, C. Papadopoulos, L. Wang, B. Zhang, Named data networking, SIGCOMM Comput. Commun. Rev. 44 (2014) 66–73.

[16] W. Shang, A. Afanasyev, L. Zhang, The design and implementation of the ndn protocol stack for riot-os, in: 2016 IEEE Globecom Workshops (GC Wkshps), pp. 1–6.

[17] A. Dowling, Y. Liu, L. Huie, K. Chen, Building an information-centric and lora-based sensing platform for iot, in: IEEE INFOCOM 2021 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 1–6.

[18] T. Ramezanifarkhani, P. Teymoori, Securing the internet of things with recursive internetwork architecture (rina), in: 2018 International Conference on Computing, Networking and Communications (ICNC), pp. 188–194.

[19] E. Grasa, M. Tarzan, L. Bergesio, B. Gastón, V. Maffione, F. Salvestrini, S. Vrijders, D. Staessens, Irati: Open source rina implementation for linux, Software Impacts 1 (2019) 100003.

[20] S. Arshad, M. A. Azam, J. Loo, M. F. Majeed, A. Haider, Hierarchical, attribute and hash-based naming and forwarding aided smart campus of things, Sustainability 15 (2023).

[21] B. Patil, D. Vydeki, Node authentication and trust-aware secure routing in ndn-iot empowered savvy city applications, Wireless Personal Communications 132 (2023) 457–485.

[22] P. Kietzmann, J. Alamos, D. Kutscher, T. C. Schmidt, M. Wählisch, Long-range icn for the iot: Exploring a lora system design, in: 2022 IFIP Networking Conference (IFIP Networking), pp. 1–9.

[23] P. Kietzmann, J. Alamos, D. Kutscher, T. C. Schmidt, M. Wählisch, Rethinking lora for the iot: An informationcentric approach, IEEE Communications Magazine 62 (2024) 34–39.

[24] S. Bilgili, A. K. Demir, S. Alam, Ifnot: An approach towards mitigating interest flooding attacks in named data networking of things, Internet of Things 25 (2024) 101076.

[25] A. M. Alwakeel, Enhancing iot performance in wireless and mobile networks through named data networking (ndn) and edge computing integration, Computer Networks 264 (2025) 111267.

[26] Y. Meng, A. B. Ahmad, Performance measurement through caching in named data networking based internet of things, IEEE Access 11 (2023) 120569–120584.

[27] W. Rafique, A. S. Hafid, S. Cherkaoui, Softcaching: A framework for caching node selection and routing in software-defined information centric internet of things, Computer Networks 235 (2023) 109966.

[28] D. Li, N. Crespi, R. Minerva, W. Liang, K.-C. Li, J. Kołodziej, Dps-iiot: Non-interactive zero-knowledge proof-inspired access control towards information-centric industrial internet of things, Computer Communications 233 (2025) 108065.

[29] R. Chiocchetti, D. Rossi, G. Rossini, ccnsim: An highly scalable ccn simulator, in: 2013 IEEE International Conference on Communications (ICC), pp. 2309–2314.

[30] M. Slabicki, G. Premsankar, M. Di Francesco, Adaptive configuration of lora networks for dense iot deployments, in: NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium, pp. 1–9.

[31] A. Augustin, J. Yi, T. Clausen, W. M. Townsley, A study of lora: Long range low power networks for the internet of things, Sensors 16 (2016).

[32] A. M. Alberti, Novagenesis: Convergent information architecture, Inatel (2012).

[33] J. A. T. Gavazza, J. C. Melo, T. B. da Silva, A. M. Alberti, P. F. Rosa, F. de Oliveira Silva ane Fábio L. Verdi, J. A. Suruagy, Future Internet Exchange Point (FIXP): Enabling Future Internet Architectures Interconnection, Springer, Cham (2020).

[34] A. Networks, Aruba 200 serie access point, Technical Datasheet (2017).

[35] B. S. Propietary, Lte-m deployment guide to basic feature set requirements, GSMA (2019).