

A Low-Cost and Infrastructure-Less LoRa Wireless Network Testbed for Cognitive Internet of Things

Ye Liu, *Member, IEEE*, Pei Tian, Carlo Alberto Boano, *Member, IEEE*, Xiaoyuan Ma, Qing Yang, *Senior Member, IEEE*, and Honggang Wang, *Fellow, IEEE*

Abstract—Low-power wide area network (LPWAN) testbeds are essential for cognitive communications and networking, as they provide practical and controlled environments for testing, validating, and advancing cognitive technologies in the cognitive Internet of Things (IoT). However, establishing extensive outdoor testbeds faces a significant challenge due to the lack of robust infrastructure, limiting testing to indoor settings or a small number of devices. This constraint prevents adequate testing of cognitive communications and networking techniques. In this article, we introduce ChirpBox: an innovative, infrastructure-free, and cost-effective LPWAN testbed that revolutionizes the utilization of LoRa nodes. Beyond their conventional role in experimentation, these nodes in ChirpBox orchestrate all operations, from disseminating firmware for testing to collecting log traces at the conclusion of each test cycle. This holistic approach is enabled by our development of an all-to-all multi-channel protocol that leverages concurrent transmissions for efficient communication across multi-hop LoRa networks. Following a detailed presentation of ChirpBox's design and implementation, we demonstrate its capabilities through a practical deployment. This evaluation offers experimental insights into the testbed's performance, illustrating its operations and highlighting its potential to advance cognitive IoT research and development.

Index Terms—Cognitive Internet of Things, Infrastructure-Less Network Testbed, Experimental Tool, LoRa.

I. INTRODUCTION

Wireless network testbeds [2] are pivotal in advancing cognitive communications and networking in the realm of cognitive Internet of Things (IoT) [3]. Cognitive technologies enable autonomous learning and decision-making, optimizing communication parameters for efficiency. These testbeds facilitate the practical implementation and assessment of cognitive functionalities [4], ensuring their effectiveness in real-world scenarios. Moreover, they address challenges posed by dynamic network conditions and empower networks to adapt to

Part of this work have been presented in [1] as a best paper nominee at the *18th International Conference on Embedded Wireless Systems and Networks (EWSN)*, 2021.

Ye Liu is with the College of Artificial Intelligence, Nanjing Agricultural University, Nanjing 210095, China (e-mail: yeliu@njau.edu.cn).

Pei Tian is with the Shanghai Advanced Research Institute, Chinese Academy of Sciences, China, as well as the University of Chinese Academy of Sciences, China (e-mail: tianpei021@gmail.com).

Carlo Alberto Boano is with the Institute of Technical Informatics, Graz University of Technology, Austria (e-mail: cboano@tugraz.at).

Xiaoyuan Ma is with SKF Group, China (e-mail: ma.xiaoyuan.mail@gmail.com).

Qing Yang is with the Department of Computer Science and Engineering, University of North Texas, Denton, TX 76205 USA (e-mail: qing.yang@unt.edu).

Honggang Wang is with the Department of Graduate Computer Science and Engineering, Katz School of Science and Health, Yeshiva University, New York, NY 10016 USA (e-mail: Honggang.wang@yu.edu).

changes in the environment, interference, or user demand [5]. The significance of testbeds extends to the development of robust and secure cognitive communication protocols [6]. They provide a controlled environment to validate the reliability and security of learning and decision-making processes, ensuring the safety and privacy of cognitive IoT networks.

On the other hand, LPWANs (such as LoRa and NarrowBand-IoT) [7]–[9] have emerged as a pivotal element of the cognitive IoT paradigm, facilitating low-power device connectivity across vast regions with cost-effective radio transceivers [10]. These technologies enable long-range communication with minimal power consumption, catalyzing the deployment of large-scale cognitive IoT systems and their innovative applications [11]. In essence, LPWANs empower a new generation of cognitive IoT [12], paving the way for a more connected, sustainable, and intelligent world.

Numerous testbeds, including FlockLab 2 [13], FIT IoT-Lab [14], LinkLab [15], and NITOS [16], enable experimentation with LoRa nodes. Some facilities are tailored specifically for testing LoRa systems [17] and conducting city-wide evaluations [18]. However, most of these testbeds are inaccessible to the public or have notable limitations. Many [14], [15] deploy nodes in indoor environments with high density, which does not accurately reflect outdoor deployment scenarios where LoRa is typically used. Consequently, wireless links in these indoor testbeds may differ significantly from real-world outdoor applications [19]. Additionally, outdoor testbeds like FlockLab 2 support only a limited number of devices, hindering large-scale testing of cognitive protocols.

Establishing outdoor testbeds for LoRa experimentation poses a significant challenge due to the complexity of setting up backbone infrastructure. This infrastructure is essential for providing power to target nodes, distributing firmware updates, synchronizing test runs, and collecting data for analysis. While indoor facilities commonly rely on wired connections like Ethernet and USB for their simplicity [20], [21], outdoor environments present unique challenges. Mounting nodes and laying cables outdoors, especially in remote areas, is difficult, and access to power outlets may be limited. Cellular technologies offer an alternative to wired connections, but they come with drawbacks such as data transmission costs and unreliable service in remote locations. In scenarios where mains power is unavailable, energy efficiency becomes crucial. This necessitates low-power communication solutions for target nodes, avoiding power-hungry modules.

To bridge the gap, this paper presents ChirpBox, a cost-effective solution that facilitates the establishment of outdoor LoRa testbeds, even in the absence of a wired infrastructure

for communication and power supply to the target nodes. Our contributions can be summarized as follows:

- ChirpBox innovatively eliminates the need for power-intensive observer nodes found in traditional IoT testbeds. Instead, it relies on battery-powered target nodes interconnected to a control node via a multi-hop network. Each target node contains two separate firmwares: one managing node activities and the other for testing purposes. The control node handles tasks such as scheduling test runs and diagnosing connectivity. ChirpBox utilizes LoRa radio for all communications, reducing costs. To manage data efficiently, it employs LoRaDisC, an advanced protocol that navigates LoRa's constraints while diagnosing node connectivity.
- We developed a ChirpBox prototype using off-the-shelf components and deployed a 21-node test setup on a university campus to showcase its capabilities. Our implementation utilizes STM32 Nucleo boards paired with SX1276 radios, a popular choice in the LoRa community. We enhanced functionality by integrating a real-time clock for precise test scheduling and a global navigation satellite system (GNSS) module for synchronized time-scaling and accurate general-purpose input/output (GPIO) event profiling. Each node, powered by rechargeable Li-ion batteries, is housed in a water-tight enclosure, ensuring flexible deployment without installation constraints.
- We leverage ChirpBox to unveil novel findings related to control node placements, the performance of LoRaWAN, offset insertion, and the environmental impact on network connectivity. To foster experimentation and collaboration, we've made our ChirpBox implementation open-source at <https://chirpbox.github.io>, providing a cost-effective and readily available solution for the community.

The rest of the paper is structured as follows. Section II provides an overview of ChirpBox's architecture. In Section III, we discuss the hardware and software components of ChirpBox, followed with the testbed management tools and APIs for firmware development in Section IV. Subsequently, Section V illustrates practical applications of the testbed through exemplary use cases. Following a review of related work in Section VI, we conclude the article in Section VII with a discussion of potential future directions.

Note that this article is an extended version of [1]. We have significantly expanded the methodology and results sections, adding design details and illustrative case studies. Specifically: (i) We have enhanced our methodology by incorporating an anti-tamper real-time clock (RTC) control, eliminating the need for additional hardware. This addition safeguards ChirpBox from unintended misoperation. Additionally, we have provided APIs to facilitate a seamless integration with external sensors. (ii) Regarding design details, the conference version outlined the LoRaDisC design, including its listen-before-talk mechanism, physical settings, and the utilization of hardware interrupts for concurrent transmissions. However, in this article, we delve deeper into the core techniques of LoRaDisC, particularly concurrent transmissions and network coding. Furthermore, we have included insights into the

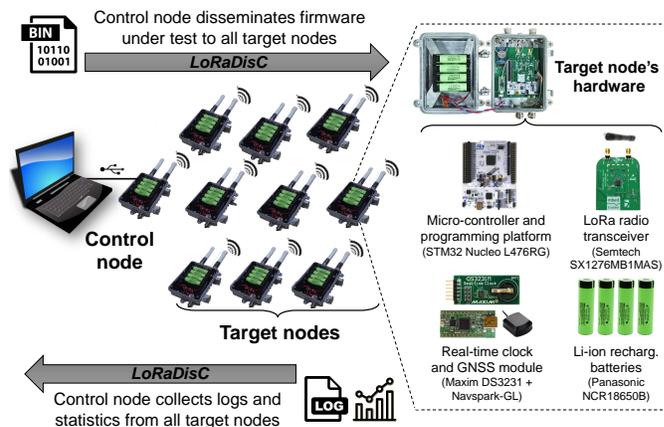


Fig. 1: The architecture of ChirpBox testbed.

health monitoring of the testbed's target nodes. (iii) We have augmented the results with a new experiment, meticulously exploring ChirpBox's performance across various control node positions. Moreover, we have provided more diverse use cases, including the impact of the LoRaWAN gateway's position on reliability. We further have verified the offset insert concept of CT-based LoRa networking on nodes deployed outdoors with ChirpBox and present a one-year observation of the performance of the LoRa network as a function of the environmental temperature.

II. CHIRPBOX IN A NUTSHELL

Fig. 1 illustrates ChirpBox's architecture, tailored for deploying LoRa-based testbeds in outdoor environments lacking communication infrastructure or power supply for target nodes. All ChirpBox nodes share identical hardware specifications, built from off-the-shelf components detailed in Section III-A. These nodes, equipped with LoRa radios, serve dual roles: executing tests and orchestrating testbed operations. ChirpBox streamlines operations by eliminating power-intensive observer nodes commonly found in traditional low-power wireless setups [20], [22]. Test runs are initiated by the user through a control node, which distributes the firmware under test (FUT) and test configurations to all target nodes. Upon completion, the control node collects all logs using LoRa communication, reducing deployment costs and enhancing flexibility by removing the need for dedicated observer nodes.

A. Control Node

ChirpBox's control node acts as the intermediary between users and the testbed, comprising a desktop PC or laptop connected via USB to a LoRa node. This node utilizes an STM32L476RG board paired with a Semtech SX1276 transceiver. Users can initiate test runs by uploading firmware (in bin file format) and specifying settings such as duration, node selection, and pre-test health checks. Inspired by D-Cube benchmarking infrastructure [23], ChirpBox features binary patching for the firmware under test, enabling real-time parameter modifications without redistributing firmware to all nodes. Test settings and protocol parameters are stored in a JSON file [24]. A Python script interprets this file

and communicates with the LoRa node via serial port for dissemination to target nodes. The script also handles result collection, including logs and testbed health status, returned to the user upon test completion.

B. LoRaDisC Protocol

Efficiently disseminating FUT and test run configurations, as well as collecting logs and health status information from battery-powered target nodes back to the control node, requires a robust dissemination/collection protocol. LoRa's limited data rates and duty cycle constraints make individual communication impractical, necessitating a multi-hop approach to enhance scalability. To address these challenges, we developed LoRaDisC, an all-to-all protocol enabling reliable and efficient multi-hop dissemination and collection across all LoRa nodes in the testbed. LoRaDisC utilizes concurrent transmissions and multiple flooding rounds to minimize collisions, maximize throughput, and accelerate information exchange while adhering to regulatory duty-cycle regulations. Additionally, LoRaDisC incorporates optimizations to support multiple spreading factors, eliminate unnecessary receptions, and enhance communication reliability.

C. Target Node's Operations

To enable seamless switching between managing testbed operations and experimentation, we utilize the dual-bank flash memory capability of the STM32L476RG microcontroller. Each target node stores two firmwares: a daemon firmware on the first memory bank handles synchronization with the control node, message transmission within LoRaDisC, FUT storage and verification, test run execution, and log trace transfer. The FUT resides on the second memory bank and can log information using ChirpBox's Application Programming Interface (API) function `log_to_flash()`. An Real-Time Clock (RTC) module ensures smooth transition from FUT to daemon once a test run ends. Further details on this process are available in Section III-B.

D. Testbed Management Features

To facilitate experimentation, ChirpBox enables users to monitor the testbed's health status and assess node connectivity. By activating the 'connectivity_evaluation' flag when initiating a test run, ChirpBox conducts a brief all-to-all communication phase to estimate link quality and packet reception rates across the network. This helps developers identify each node's neighbors and asymmetric links. During this phase, ChirpBox utilizes existing data collection primitives for log trace transmission to provide evaluation results and node health updates (e.g., battery consumption) to the user. These insights empower developers to perform critical system checks and exclude problematic nodes from tests by adjusting test run settings. Additionally, ChirpBox offers wireless daemon update capabilities and extends its API with commands for log timestamping, enhancing its functionality. We delve deeper into these features in Section IV.

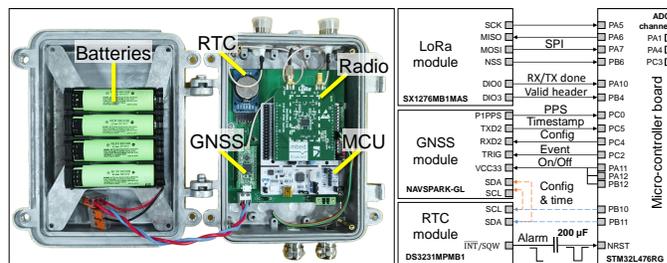


Fig. 2: Simplified diagram illustrating the connections between the primary hardware components in a ChirpBox node.

III. DESIGN AND IMPLEMENTATION

In this section, we first discuss the hardware specifications of a target node. Then, we delve into its internal operations, flash space allocation, and RTC functionalities. Finally, we provide the design of LoRaDisC and explain its role in data dissemination and collection.

A. Anatomy of a ChirpBox Node

As shown in Fig. 1, ChirpBox nodes utilize the STM32L476RG Nucleo board paired with a Semtech SX1276MB1MAS shield, operating within the 470 and 868 MHz ISM bands. This combination is widely used, including in "The Things Network" public community initiative. Each micro-controller provides a read-only unique identifier (UID), mapped by ChirpBox to a static logical ID. For example, in a testbed with nine target nodes and a control node, logical IDs range from 1 to 10. Specifying these logical IDs before compiling the daemon firmware is crucial for the built-in multi-hop protocol (LoRaDisC) to function properly.

We enhanced ChirpBox nodes by adding a GNSS module and antenna (NavSpark-GL), ensuring access to an absolute and synchronized timescale. Each node also features an RTC module (Maxim DS3231MPMB1) connected to the micro-controller's reset pin to precisely regulate test run durations. For power, each target node includes four Li-ion rechargeable batteries with a capacity of 3400 mAh, housed in a watertight IP 67 casing. The control node doesn't require batteries as it can be powered via USB. ChirpBox's setup simplicity is advantageous, requiring no additional hardware beyond the control and target nodes. This keeps overall costs low. Excluding the main board and LoRa radio, additional components are budget-friendly, costing well under \$100 per node.

Fig. 2 illustrates the connections between ChirpBox's main hardware components. The LoRa transceiver interfaces with the STM32L476RG micro-controller via SPI and two interrupt pins (DIO 0 and 3), signaling packet transmission/reception and valid header detection. Leveraging these interrupts, ChirpBox efficiently supports CT-based LoRa communication and filters out invalid headers. The micro-controller acts as a hub, retrieving time when second-level accuracy suffices, and can command the RTC module through the programmable GNSS module to generate precise alarm interrupts. This ensures accurate timing for experiments and test run conclusions. Specifically, the \overline{INT}/SQW pin is linked to the STM32L476RG's

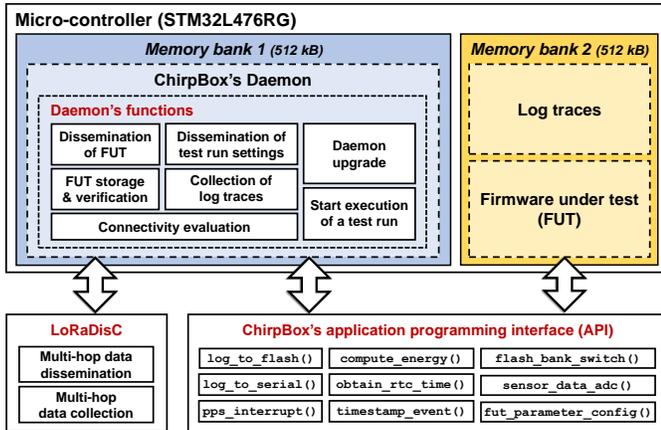


Fig. 3: Target node's components in ChirpBox testbed.

reset pin using a 200 μ F capacitor to convert falling edges into negative pulses, resetting the micro-controller.

While the RTC module could be controlled directly by the STM32L476RG micro-controller via I^2C , this increases vulnerability to misconfigurations or tampering by the FUT. To address this, our implementation restricts RTC module access solely to secure commands via the GNSS module. The STM32L476RG micro-controller directly connects to the Navspark-GL using multiple GPIO pins, enabling dynamic power management of the GNSS module to optimize energy consumption. Through the Navspark-GL, the micro-controller synchronizes its local clock with GNSS time using the PPS signal. Additionally, it accurately timestamps events using the TRIG pin and captures associated GNSS time on the PC5 pin, facilitating fine-grained debugging of distributed events.

The STM32L476RG micro-controller in a ChirpBox node features dual flash memory banks for flexible booting configurations. By adjusting the BFB2 bit at runtime, the system seamlessly switches between booting from either memory bank. This setup allows loading a daemon on the first memory bank to orchestrate the target node's activities, reserving the second bank for storing the FUT and logs related to the current test run, as depicted in Fig. 3. Before and after each test run, the micro-controller undergoes a reset, adjusting the booting configuration accordingly. The daemon, equipped with various functions detailed in Section III-B, efficiently coordinates the target node's operations using LoRaDisC's dissemination and collection primitives. ChirpBox further enhances usability with an API that simplifies logging tasks and abstracts the complexities of the GNSS and RTC modules.

B. Target Node Operations

The daemon coordinates the target node's activities following the sequential steps outlined in Fig. 4. Initially, nodes remain idle, operating in low-power mode, and periodically listen for SYNC messages broadcasted by the control node at regular 60-second intervals in our implementation. Upon receiving these messages, each target node promptly retransmits them to propagate throughout the network. SYNC messages serve dual purposes: alerting target nodes to upcoming scheduled test runs and guiding the daemon on subsequent actions

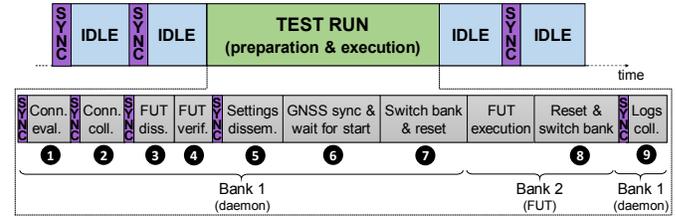


Fig. 4: Operations of a target node in ChirpBox.

required for test preparation or execution. These actions follow a predetermined sequence. Optionally, a connectivity evaluation ① may be conducted using LoRaDisC. Once completed, the control node prompts all target nodes to transmit their collected data back, also leveraging LoRaDisC ②.

The SYNC message notifies target nodes that subsequent LoRaDisC messages will include the upcoming firmware for execution ③. This firmware is seamlessly received and stored in the secondary memory bank during transmission. Following dissemination, the daemon promptly verifies the firmware's integrity ④. Notably, these steps are optional, as the test run may use previously installed firmware with modified parameters. Additionally, the control node disseminates configuration settings for the upcoming test run ⑤, including parameters like start time, duration, and FUT specifications.

If the target node is scheduled to participate in the upcoming test run, the daemon activates the GNSS module by toggling the On/Off pin. It synchronizes the target node's local clock to UTC, utilizing the PPS signal if necessary. This synchronization process takes approximately 40 seconds until the GNSS module completes its cold start phase. Then, the daemon configures the RTC module to generate alarm interrupts at the designated start and completion times of the test run. Following these preparations, the micro-controller transitions into low-power mode to conserve energy until the designated start time of the test run ⑥. Meanwhile, the daemon sets the BFB2 bit to 1 and resets the micro-controller, seamlessly executing the FUT from the secondary memory bank ⑦.

Once the designated test run duration concludes, the target node is reset by an alarm interrupt from the RTC module. Utilizing the current GNSS time, the reset handler in the STM32L476RG verifies that the RTC alarm time has passed (or waits until it does) and initiates a soft reset. Before the reset, it clears the BFB2 bit and re-enables flash write protection for bank 1 ⑧. Consequently, the target node boots from the primary memory bank and resumes ChirpBox daemon operations. The subsequent SYNC message instructs all target nodes to transmit stored logs from the previous run, located in a designated flash section, to the control node via LoRaDisC's collection primitive ⑨. Following this transmission, the target node enters low-power mode, periodically awakening to receive SYNC messages and respond appropriately.

The FUT, running on the second memory bank, must be stopped by the RTC module's alarm signal, indicating to the micro-controller that a test run is complete. This is crucial for returning control to the daemon. If the RTC module fails to reliably trigger the micro-controller at the end of a test run, a

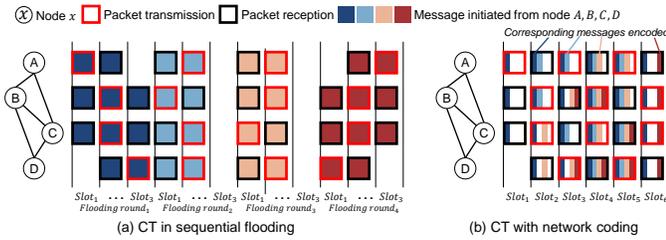


Fig. 5: Example of data dissemination.

target node might not switch back to the daemon firmware. To mitigate this risk without additional hardware, we connect the RTC module to the GNSS module's I^2C interfaces instead of directly to the micro-controller, as depicted in Fig. 2. Through the easily-programmable interface provided by the GNSS module, we implement a set of private commands known only to the daemon to configure and fetch the time/alarm time of the RTC module. This enables the micro-controller in the daemon to control the RTC module, keeping it hidden from the FUT.

C. LoRaDisC

ChirpBox utilizes LoRaDisC, a versatile multi-hop protocol facilitating both one-to-all data dissemination and all-to-one data collection across LoRa nodes. LoRaDisC harnesses concurrent transmissions (CT), maintaining their simplicity and advantages. These include low end-to-end latency, exceptional reliability, seamless multi-hop communication without complex routing strategies, and remarkable energy efficiency [25].

1) *Concurrent Transmissions*: The transmission of multiple overlapping packets over the air typically results in a collision, and a wireless device is often unable to decode a packet correctly. However, it was recently shown that by means of concurrent transmissions, i.e., by letting devices transmit their packets simultaneously and by accurately aligning the start of the transmissions, there is a high chance to correctly decode packets [25]. Specifically, a target packet has a high chance to be decoded correctly if it is received with a greater power (>3 dB) than the sum of all the other signals at the receiver: this effect is called capture effect [26]), and it has been analyzed theoretically [27], [28] and verified experimentally on several off-the-shelf transceivers [29].

CT have been employed to achieve reliable data collection and dissemination [30]. In a basic CT-based network, multiple sequential flooding rounds are conducted as shown in Fig. 5(a). In each round, a node initiates the flooding process by actively transmitting a packet, which is then rebroadcasted by other nodes in subsequent CT slots. Each node is limited to transmitting at most N_T times per round to conserve energy, with each round comprising N_S slots ($N_T = 1$ and $N_S = 3$ in Fig. 5(a)). This approach ensures message dissemination to all nodes in the network. Initially implemented using IEEE 802.15.4 narrowband transceivers [25], [29], CT feasibility has been confirmed on various platforms, including IEEE 802.15.4 ultra-wideband radios [31] and Bluetooth Low Energy (BLE) devices [32]. In the context of LoRa, CT feasibility was investigated in [33] and further validated in [34].

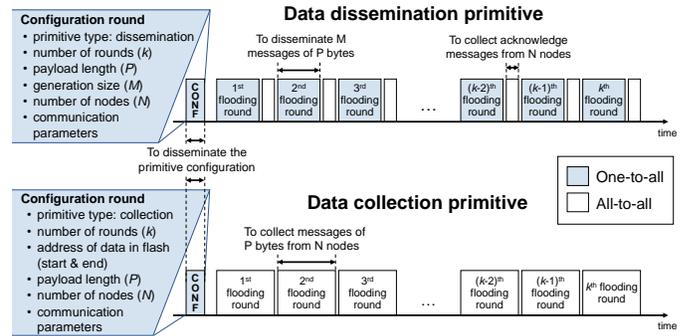


Fig. 6: LoRaDisC facilitates two communication primitives.

2) *Network Coding*: Network coding has been utilized in CT-based networks to enhance their throughput and reliability [35], [36]. Network coding involves delivering encoded packets to reduce redundancy (e.g., using fountain codes [37]) and optimize packet exchange (e.g., Luby Transform (LT) codes [38] and random linear network coding (RLNC) [39]). RLNC is an example of network coding applied to CT networks, as depicted in Fig. 5(b). By employing network coding, the required number of CT slots for data dissemination decreases from 12 (as shown in Fig. 5(a)) to 6. Unlike sequential CT flooding, RLNC-encoded packets are not identical. Instead, colorful blocks in Fig. 5(b) represent encoded messages, such as those obtained through XOR operations. Receivers decode packets based on known messages. For example, in Fig. 5(b), node C can decode B's message during the third CT slot because it has already decoded A's message in slot 1.

3) *Communication Primitives*: LoRaDisC comprises a sequence of flooding rounds, commencing with a one-to-all round that conveys configuration details for the subsequent rounds, as depicted in Fig. 6. This includes the receiving nodes of several parameters: (i) the primitive type, whether dissemination or collection; (ii) the number of messages M to disseminate or the flash start/end addresses for data to collect; (iii) the payload length P for subsequent messages; (iv) the number of following flooding rounds k ; (v) the network's node count N ; and (vi) LoRa communication specifics such as SF, bandwidth, transmission power, and coding rate for use in later rounds. Each flooding round employs CT slots, blending packets using RLNC for reliability and higher throughput. Coded packets in LoRaDisC enhance successful deliveries over unreliable channels. Following the initial one-to-all round, an all-to-all round ensures all nodes receive the preceding data, acknowledged by each node setting a bit in the LoRaDisC header's coding vector field. If not all bits are set, retransmission occurs in the next round.

4) *Use of LBT and AFA*: LoRaDisC's design is heavily influenced by regulations governing spectrum access for LoRa systems. These regulations impose strict limits on transmission frequency. In Europe, LoRa devices must adhere to a transmission duty cycle as low as 0.1% or 1%, depending on the channel, without employing polite spectrum access techniques. This translates to maximum transmission times of 3.6 seconds or 36 seconds per hour per channel. However, incorporating listen-before-talk (LBT) and adaptive frequency

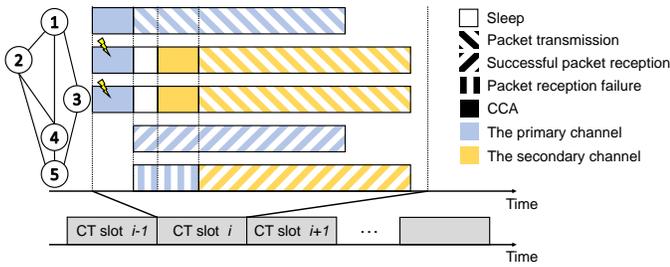


Fig. 7: LBT & AFA mechanisms in LoRaDisC.

agility (AFA) can significantly relax these constraints. LBT requires devices to perform a clear channel assessment (CCA) and wait for a clear channel or switch frequencies if occupied. AFA utilizes multiple frequencies for transmission. With both LBT and AFA, the duty cycle restriction allows up to 100 seconds of cumulative transmission time per channel per hour, corresponding to a duty cycle ratio of 2.7%.

In LoRaDisC, we've integrated listen-before-talk and adaptive frequency agility mechanisms to increase the duty cycle per channel for faster data transfer, as shown in Fig. 7. To comply with spectrum access regulations, each node undergoes a 5-millisecond listening period before transmission to ensure channel clarity. This approach differs from traditional Carrier Sense Multiple Access (CSMA) protocols, where messages can be promptly retransmitted. However, in LoRaDisC, nodes conduct a CCA check first. Fortunately, LoRa's low data rate tolerates the additional LBT delay, making it suitable for our purposes compared to IEEE 802.15.4 standards.

In each CT slot, nodes utilize both primary and secondary channels, as shown in Fig. 7. At the slot's start, nodes intending to transmit (e.g., nodes 1, 2, and 3) perform a CCA check on the primary channel. If clear, transmission begins after a 5 ms delay (node 1). If the primary channel is busy due to RF activity (nodes 2 and 3), the node waits, performs a CCA check on the secondary channel, and transmits if clear. Receivers (nodes 4 and 5) primarily listen on the primary channel. If data is detected, they receive it (node 4). If not, they switch to the secondary channel (node 5). Channels are dynamically adjusted for maximum throughput and compliance with regulations. Nodes select channels based on round and slot numbers, with slot duration calculated post-configuration, considering factors like payload length, spreading factor, bandwidth, and coding rate.

5) *Physical Layer (PHY) Settings*: During LoRaDisC's initial all-to-all configuration round, predetermined PHY settings are used, but subsequent rounds can adjust these settings based on exchanged data. Since parameters like spreading factor (SF) significantly impact frame airtime, affecting transmission frequency, LoRaDisC dynamically adjusts maximum payload length to meet local regulations. While LoRa supports payloads up to 255 bytes, local limits may impose restrictions. Thus, before using LoRaDisC, the transmitter (e.g., ChirpBox's control node) calculates payload size based on the SF for the upcoming round. In our implementation, with LoRaDisC header and footer totaling 14 bytes for 20 nodes, only SFs up to 11 are supported in Europe.

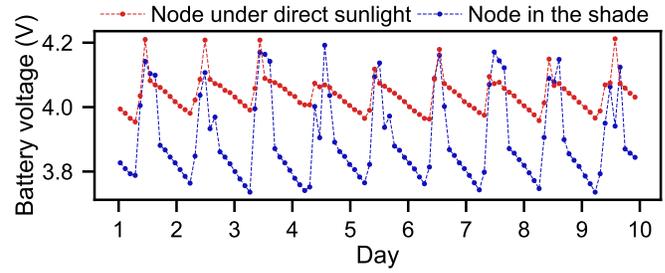


Fig. 8: Evolution of the battery voltage on ChirpBox nodes powered by solar panel.

6) *Use of Hardware Interrupts*: Numerous protocols, including those based on IEEE 802.15.4 radios [25] and BLE transceivers [40], have recently emerged utilizing CT. However, implementing CT on LoRa faces challenges due to the lack of a start-of-frame delimiter (SFD) interrupt signal, crucial for node synchronization [41]. Since most LoRa radios, like the SX1276 in ChirpBox, lack an SFD interrupt, we rely on RX done and TX done interrupts (DIO 0) instead. Despite their jitters (1.48 ms and 43.6 μ s respectively [42]), they're tolerable within a LoRa CT implementation, accommodating alignment errors of up to 3 symbol times (3.072 ms for SF=7) [34]. To boost LoRaDisC's efficiency, we employ a second interrupt (DIO 3) triggered upon receiving a valid header. Given LoRa's low data rate, nodes promptly verify if the next byte matches the LoRaDisC header to conserve energy by powering off the radio if not. This aids in filtering transmissions from co-located LPWANs.

IV. TESTBED SERVICES

In this section, we present ChirpBox's multifaceted functionalities for monitoring the testbed's health and connectivity, along with facilitating firmware upgrades and patch file generation. An API allows the FUT to timestamp events and store log files, while support for external sensors, like acoustic sensors, enables flexible application extensions.

A. Services for Testbed Management

Control node can instruct target nodes to periodically collect and transmit data, offering insights into the testbed's health and connectivity. This includes monitoring battery voltage and onboard temperature, with nodes relaying this information back to the control node. Analyzing voltage levels enables precise identification of nodes requiring battery replacements.

1) *Health Monitor*: The battery voltage serves as a critical health indicator in ChirpBox, collected periodically from target nodes. If the voltage drops below a certain threshold, a node may malfunction due to unexpected memory bank switches during resets, potentially interrupting test runs. To address this, we implement an under-voltage protection mode in the daemon firmware. Upon detecting low voltage, the daemon enters this mode, where it continuously monitors the voltage every second until it stabilizes above the start-up threshold, ensuring uninterrupted operation.

ChirpBox is compatible with solar panel power, offering a sustainable energy solution. With a 12-to-3.3 V adapter,

a ChirpBox node can be integrated into existing solar street lamp posts, leveraging solar energy for operation (Fig. 9(b)). We assess battery-powered nodes equipped with solar panels for charging. Our evaluation includes connectivity evaluations (phase 1) at two-hour intervals on three channels spanning SF 7 to SF 12 over a period of up to 10 days, with an average of 10.3 hours of daylight each day. The results in Fig. 8 show that despite shading challenges and limited exposure to direct sunlight, the nodes maintain a consistently high battery voltage. This robust performance ensures stable and uninterrupted operation solely sustained by solar energy, rendering ChirpBox practically maintenance-free by eliminating the need for periodic battery replacements.

Additionally, the on-board temperature serves as another health indicator, detecting hardware anomalies like short circuits. Utilizing the built-in sensor of the SX1276 transceiver, temperature measurements are obtained. In Section V-D, we demonstrate how these readings reflect ambient temperature effects on link quality. Unlike battery voltage, temperature values are included in connectivity status information.

2) *Connectivity Monitor*: In ChirpBox, each node must have at least one neighbor to ensure information relay between nodes and the control node. Therefore, ensuring network connectivity is crucial, which can be achieved through careful node deployment. To facilitate this, ChirpBox includes a tool for measuring network connectivity, aiding in analyzing link quality between nodes. The connectivity evaluation firmware directs nodes to periodically transmit probe packets in a collision-free manner. By adjusting firmware settings, such as SFs, one can observe how network topology evolves. Each node updates packet reception ratio (PRR) statistics, persistently stored in flash memory. After evaluation, the control node gathers statistics using LoRaDisC's collection primitive. Notably, ambient temperature affects link quality, as studies suggest. Therefore, on-board temperature measurements are included in data collected by the control node, facilitating investigation of environmental impacts on LoRa communication.

3) *Daemon Upgrade*: To address bugs, add new features, or modify LoRaDisC's configuration, updating the daemon firmware is crucial. ChirpBox simplifies this process by using *jojodiff* [43] to generate a patch file, accessible to the control node. Upon activation of the patching flag in a SYNC message, the control node disseminates the patch file, along with MD5 verification. Once received, ChirpBox's daemon creates a patched image in the secondary memory bank and verifies its integrity with *janpatch* [44]. If valid, the daemon switches to the second bank, similar to executing a FUT. During runtime, the patched daemon checks the SYSCFG_MEMRMP register to determine the current bank. If on bank 2, it duplicates itself to bank 1. ChirpBox then resumes execution of the newly patched daemon from bank 1, following the FUT execution steps. Notably, the same patching tools can be used for FUTs, particularly when firmware differences are minimal. This approach significantly reduces the time required for FUT dissemination by the control node.

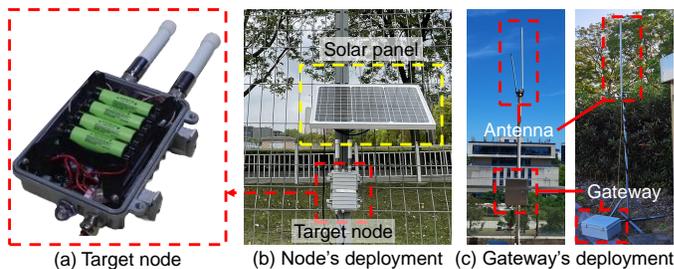


Fig. 9: Example of ChirpBox target node.

B. Application Programming Interface (API)

1) *Basic API*: ChirpBox offers developers a valuable tool for investigating and debugging protocol performance by enabling precise timestamped logging directly to flash memory, overcoming the lack of direct observer connection to target nodes. Through a low-level API, ChirpBox abstracts the complexities of underlying RTC and GNSS modules, allowing easy timestamping of GPIO events on a millisecond scale using a 6-byte Unix timestamp with a simple function call of the `timestamp_event()` function. Furthermore, the `log_to_flash()` function seamlessly records requested information alongside timestamps directly to flash memory, optimizing memory usage by storing formatted strings and variables separately. For example, a 310 kB log file generated from a 3-hour test involving 21 nodes occupies only about 3.75 kB of flash memory. After completing a test run, the control node automatically collects these stored log messages, facilitating convenient conversion to readable format via a Python script on a connected laptop. ChirpBox also supports on-site inspections, enabling logs to be redirected and conveniently outputted via the serial port using the `log_to_serial()` function.

2) *APIs for External Sensors*: To enhance ChirpBox's functionality, users can integrate additional sensors into its target nodes. For instance, connecting a microphone to a node allows real-time reading of sound levels using the API's `sensor_data_adc()` function. This enables the generation of event-based traffic, mimicking real-world application patterns more accurately than periodic traffic. Additionally, nodes can serve as low-power data acquisition devices, recording real-world data such as sound levels [45] locally and transmitting it to the control node after a test run concludes.

V. CHIRPBOX IN ACTION

In our previous study [1], we conducted a thorough assessment of the LoRaDisC protocol's performance across various parameters including file size, spreading factor, and network size. Additionally, we examined the overhead incurred by ChirpBox in orchestrating the operations of the testbed and analyzed the energy consumption breakdown during each phase of experiment preparation, execution, and conclusion.

In this section, we leverage ChirpBox to unveil novel insights pertaining to the placement of control nodes, the performance of LoRaWAN, the concept of offset insertion, and the impact of environmental factors on network connectivity. Furthermore, a benchmarking analysis of protocol performance is carried out. The ChirpBox testbed includes

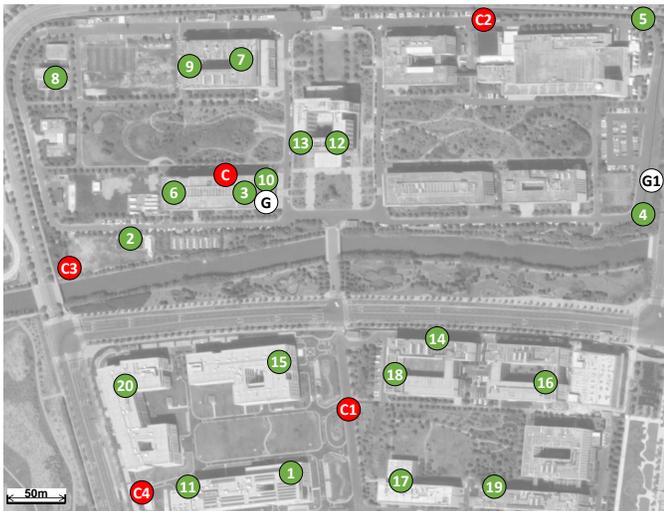


Fig. 10: ChirpBox test setup on a university campus, comprising twenty target nodes (marked in green) and a single control node (highlighted in red).

21 nodes across a 28-hectare area on a university campus in Shanghai (see Fig. 10). Fig. 9(a) displays a typical ChirpBox target node, while Fig. 9(b) shows a solar-powered variant. The deployment of two gateways is depicted in Fig. 9(c).

A. Control Node at Different Locations

We evaluate ChirpBox’s performance by collecting 2 kB logs and disseminating 50 kB firmware updates (FUTs) from five different positions of the control node, as marked in Fig. 10. LoRaDisC consistently achieves 100% reliability regardless of the control node’s position. However, there are slight variations in collection/dissemination latency and overall energy consumption across different positions.

Fig. 11a illustrates the results of data collection. The lowest average energy consumption and shortest duration occur when the control node is positioned at location C1 (23.3% less energy than at location C), while the highest energy consumption is observed at location C4 (2.7% more than at location C). Utilizing ChirpBox’s connectivity monitor, we observe that location C1 has the most neighbors, facilitating easier data collection. Conversely, location C4 with the fewest neighbors experiences higher energy consumption. Fig. 11b displays the results of data dissemination. Optimal performance occurs when the control node is at location C2 (21.4% less energy than at location C), while the least efficient scenario is observed at location C4 (6.4% more energy than at location C). Utilizing ChirpBox’s connectivity monitor, we ascertain that nodes 4 and 5 have weak links with the network, requiring more attempts to send acknowledgments. Hence, positioning the control node at location C2 yields the best performance due to the higher number of neighboring target nodes.

B. Investigating LoRaWAN’s Performance

We deploy LoRaWAN gateways at locations G and G1, respectively, as marked in Fig. 10, enabling target nodes to periodically send packets using LoRaWAN Class A. The

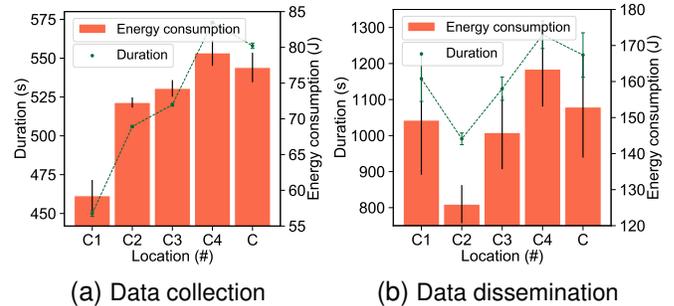


Fig. 11: LoRaDisC’s performance when placing the control node at different locations.

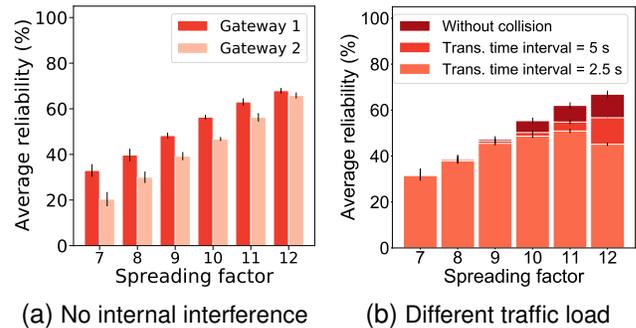
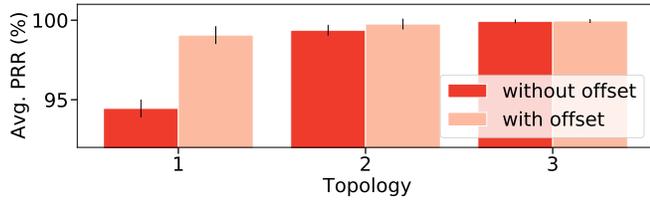


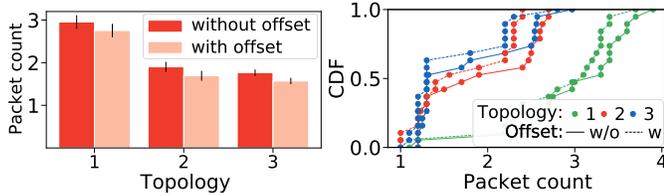
Fig. 12: Reliability of LoRaWAN communications when using different gateway locations and with different traffic loads.

gateways listen to eight channels, and ChirpBox’s target nodes upload packets on a random channel from these eight. We assess communication reliability by conducting tests with a transmission power of 0 dBm and SFs ranging from 7 to 12. To investigate how collisions affect the reliability of this ALOHA network, we ensure the absence of internal interference by letting target nodes send packets in a round-robin manner, utilizing the GNSS module of ChirpBox as a reference. After each test, results would be collected by using LoRaDisC (i.e., the Logs coll. depicted in Fig. 4). It is worth noting that the LoRaDisC protocol is primarily responsible for disseminating the firmware image prior to testing and then collecting the performance results once the test is complete. Therefore, during the actual evaluation of the LoRaWAN protocol, the LoRaDisC protocol remains inactive, thus ensuring that there is no interference with the performance evaluation.

Fig. 12a displays the average Packet Reception Ratio (PRR) in collision-free tests with the gateway positioned at locations G and G1. Higher spreading factors enhance coverage at the expense of lower data rates, significantly boosting reliability and energy consumption (e.g., PRR at SF=12 doubles, but energy consumption rises by about 26 times compared to SF=7). Interestingly, the gateway’s position becomes less critical with higher SFs; G1’s reliability slightly lags behind G at lower SFs but nearly matches it at higher SFs. In Fig. 12b, we examine LoRaWAN reliability under various traffic loads, uploading data every 5 s and 2.5 s, respectively, with the gateway at location G. Here, the slow data rate constrains throughput,



(a) Reliability as a function of the chosen topology



(b) Avg. first received packet (c) CDF of first received packet

Fig. 13: Evaluating the impact of the offset insert concept.

but the average PRR remains stable, dropping marginally with more frequent transmissions at SF=12.

C. Investigating the “Offset Insert” Concept

Liao et al. [34] verified the feasibility of CT over LoRa, and improved its reliability by inserting an additional time offset in each CT slot. The latter helps spreading the energy of interference (i.e., signals from other CT nodes) in the time domain, thereby ensuring that packets are decoded successfully.

We verify this *offset insert* concept on the ChirpBox network deployed presented before. In order to make our study topology-independent, we set nodes 5, 11, and 8 as flooding initiators and denote these three configurations as topologies 1–3, respectively. The initiator triggers the network by actively sending a packet, whereas the remaining nodes simply re-broadcast received packets immediately, following an *rx-tx-rx-tx* pattern. Each node has 5 transmission attempts during each flooding round. We also set SF=7 in order to create a larger multi-hop network. The logged stats contain PRR information as well as an indication of the first packet being received.

Fig. 13a shows the reliability in terms of average PRR, one can notice that the position of the flood initiator does affect performance. Specifically, the reliability degrades as the network diameter increases (i.e., the average PRR is only 94% in topology 1 with the conventional CT flooding). However, by using the *offset insert*, the reliability of topology 1 is improved significantly, with the PRR approaching 100%. With the help of the logged information about the first packet being received, we can show that the *offset insert* really increases the chances to decode a CT packet correctly, thereby decreasing the necessary number of packet retransmissions. Fig. 13b and Fig. 13c confirm that the first packet being received is in average lower when applying the offset in all three topologies.

D. Observing Environmental Impact on LoRa Networks

By utilizing the ChirpBox-collected testbed health status information, as discussed in Section IV-A2, users can effortlessly gather data on the on-board temperature of each

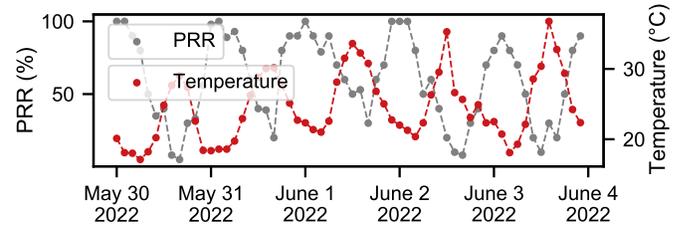


Fig. 14: Impact of ambient temperature on the PRR between node 7 and 3. We use 480 MHz as channel and SF=7.

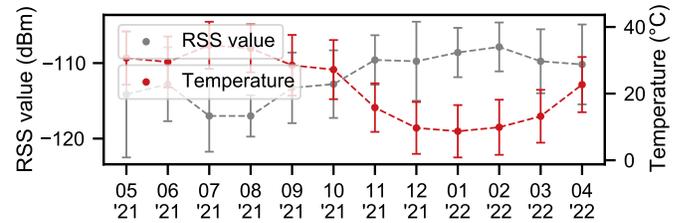


Fig. 15: Impact of ambient temperature on the Received Signal Strength (RSS) values recorded by node 1 over a 12-month period (May 2021 to April 2022).

target node over extended periods. This information can then be used to examine whether temperature fluctuations have any discernible effects on the connectivity within the testbed. By powering these nodes deployed in the university campus by mains or using solar panels, we record the connectivity information as a function of temperature over more than one year (from May 2021 to June 2022). Fig. 14 shows the exemplary link quality between node 7 and node 3. The PRR measured at node 3 fluctuates periodically, decreasing significantly when the temperature rises (i.e., during daytime) and recovering when the temperature decreases (i.e., during nighttime). Fig. 15 depicts the Received Signal Strength (RSS) recorded by node 1 while receiving packets from node 20 over a span of 12 months (May 2021 to April 2022) using a channel frequency of 480 MHz and employing Spreading Factor (SF) 7. The observed RSS values exhibit a distinct correlation with temperature, displaying lower values during the summer (characterized by higher temperatures) and higher values in the winter (associated with lower temperatures). This example confirms earlier studies [46], [47] and indicates that higher temperatures negatively affects LoRa communication. Moreover, a comprehensive evaluation of the environmental impact on the long-term connectivity and link quality of outdoor LoRa networks, along with a corresponding dataset, was presented in [48], utilizing our ChirpBox testbed. We are confident that the availability of ChirpBox will not only foster experimentation with LoRa technology, but also significantly contribute to the development of cognitive communications and networking protocols in the future.

E. Benchmarking Protocol Performance

Finally, we compared three LoRa-based protocols: LoRaDisC with LBT and AFA, LoRaBlink, and LoRaWAN.

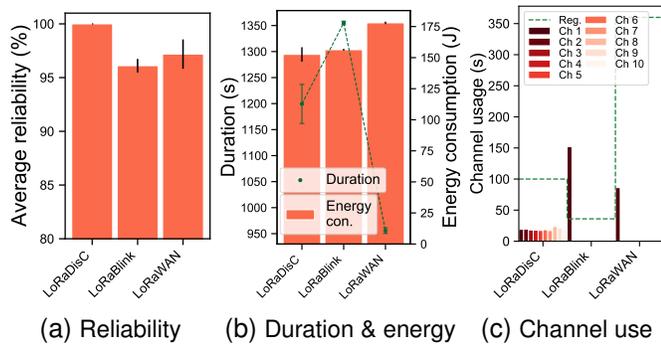


Fig. 16: Performance of LoRaDisC, LoRaBlink, and LoRaWAN when disseminating a 50 kB file across a network.

Our evaluation focused on end-to-end latency, energy consumption, and channel usage during the dissemination of a 50 kB firmware image. For LoRaDisC, we maintained the settings described earlier, including the use of LBT allowing transmissions of up to 100 seconds per hour per channel. LoRaBlink [33], a multi-hop protocol, was assessed using its publicly available firmware. In our tests, the control node C sent beacon packets every 7.5 seconds (epoch) to maintain synchronization among nodes. Notably, LoRaBlink lacks an LBT mechanism, limiting data transmission to a maximum of 36 seconds per hour per channel.

LoRaDisC and LoRaBlink operate in multi-hop network configurations, while LoRaWAN forms a star network centered around the gateway G shown in Fig. 10. LoRaWAN employs its firmware update over the air (FUOTA) process, delivering firmware to multiple end-devices in a multicast group using LoRaWAN Class C protocol. This enables low-latency communication, with end-devices listening continuously except during transmission. The gateway operates at a 10% duty cycle, allowing transmissions of up to 360 seconds per hour. For a fair comparison, we focused on firmware chunk dissemination, excluding the digital signature procedure.

Fig. 16 depicts the performance of the three protocols across three distinct runs. LoRaDisC achieves 100% reliability (Fig. 16a) with the lowest energy consumption (Fig. 16b) while maintaining channel usage significantly below regulatory limits (Fig. 16c). LoRaWAN Class C emerges as the fastest protocol for dissemination, completing the task 20% faster than LoRaDisC. However, despite its shorter dissemination duration, it consumes 15% more energy than LoRaDisC due to continuous listening. Implementing a LoRaWAN Class A firmware would result in a 6.6-fold increase in duration and a 3.4-fold rise in energy consumption to comply with regulations. Conversely, LoRaBlink exhibits the lowest reliability and longest dissemination latency, with comparable energy expenditure to LoRaDisC. This is despite utilizing the channel approximately four times more than permitted by regulations (Fig. 16c). If LoRaBlink adhered to channel usage regulations, its dissemination duration would increase by 6.6 times.

VI. RELATED WORK

We now delve into related works concerning existing LoRa-based testbed facilities and concurrent transmission (CT) protocols utilizing LoRa technology.

1) *LoRa-based Testbed Infrastructures*: Numerous purpose-built testbeds have been developed to evaluate the reliability and scalability of LoRa systems, as evidenced by studies [15]–[18], [49]–[54]. However, access to most of these facilities is restricted to specific users. Recently, several public testbed infrastructures, such as FlockLab 2 [13], [55], FIT IoT-Lab [14], and UMBRELLA [56], have emerged to support LoRa devices. Nevertheless, these platforms often feature a limited number of indoor nodes. To simplify the reprogramming and management of target LoRa nodes, these facilities rely on existing network backbones, typically Ethernet or cellular networks [56]–[58]. Acknowledging the limitations of network backbones and the potential operational costs associated with cellular networks, Kazdaridis et al. [16] have chosen to exclusively utilize LoRa nodes in their experiments. Their methodology involves employing LoRaWAN gateways to transmit configuration commands to target nodes and gather statistics. However, this approach requires manual firmware programming before deployment and supports only star topologies, constraining scalability. In contrast, ChirpBox introduces a comprehensive multi-hop testbed that enables reprogramming of LoRa nodes and efficient log trace collection. In recent times, a comprehensive survey has been conducted to thoroughly review the current state-of-the-art research on LoRa technology, particularly emphasizing its testing and evaluation methodologies [59]. This survey encompasses a broad range of models, simulators, and testbeds that have been developed to assess the performance and capabilities of LoRa-based systems. By analyzing these various approaches, readers can gain a deeper understanding of the current trends and challenges in evaluating LoRa systems, paving the way for future research and improvements in this field.

2) *Concurrent Transmissions on top of LoRa*: CT has gained widespread popularity within low-power wireless systems, tracing its origins to the pioneering work on Glossy [29]. This significant contribution has spurred the development of various CT-driven data collection and dissemination protocols tailored for IEEE 802.15.4 systems [25]. Recent studies have showcased the feasibility of CT across diverse technologies, encompassing Bluetooth Low Energy [60], ultrawideband [61], and LoRa [62]. Notably, in the realm of LoRa, Bor et al. [33] were the first to empirically verify the existence of non-destructive CTs under specific conditions. These insights were further expanded upon by Liao et al. [34], who conducted a theoretical analysis and developed a prototype implementation of CT atop LoRa. Our work builds upon these two studies and proposes, to the best of our knowledge, the very first CT-based multihop data collection and dissemination protocol for LoRa-based networks. The proposed LoRaDisC protocol copes with LoRa's limited data rate and regional duty-cycle constraints and allows ChirpBox to sustain a reliable and efficient data exchange across the

testbed. A comprehensive survey has been undertaken [25], centering on communication protocols and network services that leverage concurrent transmission techniques. This survey not only delves into the intricacies of these technologies, but also emphasizes the distinguishing features and commonalities among the various proposed solutions. By comparing and contrasting these approaches, readers can obtain a nuanced understanding of the current landscape in this field.

VII. CONCLUSIONS AND FUTURE WORK

We have presented ChirpBox, a low-cost and infrastructureless LoRa testbed that can be deployed in remote areas lacking cellular coverage and backbone infrastructure. ChirpBox facilitates efficient communication with target nodes and provides power supply without reliance on external infrastructure. Leveraging LoRaDisC, the first CT-based all-to-all multi-channel protocol on top of LoRa, ChirpBox achieves reliable and efficient data dissemination and collection across the network. This capability enables seamless orchestration of testbed activities by utilizing LoRa-based target nodes. We believe that the accessibility and cost-effectiveness of ChirpBox, coupled with its open-source nature, will foster research on cognitive communications and networking.

Moving forward, we anticipate the need for aperiodic traffic generation in order to emulate event-driven applications, which can be achieved with the help of the programmable GNSS module. To support the addition or removal of target nodes after deployment, the built-in protocol, LoRaDisC, needs to periodically form a network (i.e., it should update the online targets). Future work on ChirpBox will also focus on enhancing scalability and integrating a centralized cloud service. This involves optimizing algorithms for larger networks, implementing hierarchical management, and refining performance. A common, openly-accessible cloud service for multiple ChirpBox instances will offer unified management, scalability, and advanced analytics. Additionally, security measures, user experience improvements, and customization options will be prioritized to ensure ChirpBox remains robust, user-friendly, and capable of meeting the demands of large-scale network deployments.

ACKNOWLEDGMENTS

We would like to express our gratitude to Fengxu Yang, Xin Tian, Dan Li, and Sixuan Chen for their invaluable contributions in constructing the ChirpBox testbed.

REFERENCES

- [1] P. Tian, X. Ma, C. A. Boano, Y. Liu, F. Yang, X. Tian, D. Li, and J. Wei, "ChirpBox: An Infrastructure-Less LoRa Testbed," in *Proceedings of the 2021 International Conference on Embedded Wireless Systems and Networks*, ser. EWSN '21. USA: Junction Publishing, 2021, p. 115–126.
- [2] C.-H. Kuo, G.-C. Yang, D. Chacko, and W. C. Kwong, "True-Asynchronous Transmission Modeling and Software-Defined-Radio Experimental Testbed for Cognitive Radio Ad-Hoc Wireless Networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 10, no. 1, pp. 35–47, 2024.
- [3] J. Sun, J. Chen, G. Ding, F. Lin, and Y. Song, "Spectrum Recommendation in Cognitive Internet of Things: A Knowledge-Graph-Based Framework," *IEEE Transactions on Cognitive Communications and Networking*, vol. 10, no. 1, pp. 21–34, 2024.

- [4] G. Kakkavas, K. Tsitseklis, V. Karyotis, and S. Papavassiliou, "A Software Defined Radio Cross-Layer Resource Allocation Approach for Cognitive Radio Networks: From Theory to Practice," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 2, pp. 740–755, 2020.
- [5] J. Chen, N. Skatchkovsky, and O. Simeone, "Neuromorphic Wireless Cognition: Event-Driven Semantic Communications for Remote Inference," *IEEE Transactions on Cognitive Communications and Networking*, vol. 9, no. 2, pp. 252–265, 2023.
- [6] M. Li, K. Xue, W. Chen, and Z. Han, "Secure Performance of RIS-Aided NOMA in Cognitive V2X Networks With Imperfect CSI over Double Rayleigh Fading," *IEEE Transactions on Cognitive Communications and Networking*, pp. 1–1, 2024.
- [7] O. J. Pandey, T. Yuvaraj, J. K. Paul, H. H. Nguyen, K. Gundepudi, and M. K. Shukla, "Improving Energy Efficiency and QoS of LPWANs for IoT Using Q-Learning Based Data Routing," *IEEE Transactions on Cognitive Communications and Networking*, vol. 8, no. 1, pp. 365–379, 2022.
- [8] M. Chen, Y. Miao, X. Jian, X. Wang, and I. Humar, "Cognitive-LPWAN: Towards Intelligent Wireless Services in Hybrid Low Power Wide Area Networks," *IEEE Transactions on Green Communications and Networking*, vol. 3, no. 2, pp. 409–417, 2019.
- [9] L. M. Figueiredo and E. Franco Silva, "Cognitive-LoRa: Adaptation-Aware of the Physical Layer in LoRa-based Networks," in *2020 IEEE Symposium on Computers and Communications (ISCC)*, 2020, pp. 1–6.
- [10] U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low Power Wide Area Networks: An Overview," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 855–873, 2017.
- [11] A. Pagano, D. Croce, I. Tinnirello, and G. Vitale, "A Survey on LoRa for Smart Agriculture: Current Trends and Future Perspectives," *IEEE Internet of Things Journal*, vol. 10, no. 4, pp. 3664–3679, 2023.
- [12] K. Ntshabele, B. Isong, and A. M. Abu-Mahfouz, "CR-LPWAN: Issues, Solutions and Research Directions," in *2021 IEEE World AI IoT Congress (AIoT)*, 2021, pp. 0504–0511.
- [13] R. Trüb, R. Da Forno, L. Daschinger, A. Biri, J. Beutel, and L. Thiele, "Non-Intrusive Distributed Tracing of Wireless IoT Devices with the FlockLab 2 Testbed," *ACM Transactions on Internet of Things*, vol. 3, no. 1, oct 2021.
- [14] C. Adjih, E. Baccelli, E. Fleury, G. Harter, N. Mitton, T. Noel, R. Pissard-Gibollet, F. Saint-Marcel, G. Schreiner, J. Vandaele, and T. Watteyne, "FIT IoT-LAB: A Large Scale Open Experimental IoT Testbed," in *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, 2015, pp. 459–464.
- [15] Y. Gao, J. Zhang, G. Guan, and W. Dong, "LinkLab: A Scalable and Heterogeneous Testbed for Remotely Developing and Experimenting IoT Applications," in *2020 IEEE/ACM Fifth International Conference on Internet-of-Things Design and Implementation (IoTDI)*, 2020, pp. 176–188.
- [16] G. Kazaridis, S. Keranidis, P. Symeonidis, P. Tzimotoudis, I. Zographopoulos, P. Skrimponis, and T. Korakis, "Evaluation of LoRa Performance in a City-wide Testbed: Experimentation Insights and Findings," in *Proceedings of the 13th International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization*, ser. WiNTECH '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 29–36.
- [17] J. M. Marais, R. Malekian, and A. M. Abu-Mahfouz, "Evaluating the LoRaWAN Protocol Using a Permanent Outdoor Testbed," *IEEE Sensors Journal*, vol. 19, no. 12, pp. 4726–4733, 2019.
- [18] J. Struye, B. Braem, S. Latré, and J. Marquez-Barja, "The CityLab Testbed — Large-scale Multi-technology Wireless Experimentation in a City Environment: Neural Network-based Interference Prediction in a Smart City," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2018, pp. 529–534.
- [19] C. Hakkenberg, "Experimental Evaluation of LoRa(WAN) in Indoor and Outdoor Environments," Master's thesis, University of Twente, 2016.
- [20] M. Schüb, C. A. Boano, M. Weber, and K. Römer, "A Competition to Push the Dependability of Low-Power Wireless Protocols to the Edge," in *Proceedings of the 2017 International Conference on Embedded Wireless Systems and Networks*, ser. EWSN '17. USA: Junction Publishing, 2017, p. 54–65.
- [21] P. Appavoo, E. K. William, M. C. Chan, and M. Mohammad, "Indriya2: A Heterogeneous Wireless Sensor Network (WSN) Testbed," in *Proceedings of the 13th International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TridentCom)*, 2018.
- [22] R. Lim, F. Ferrari, M. Zimmerling, C. Walser, P. Sommer, and J. Beutel, "FlockLab: A Testbed for Distributed, Synchronized Tracing and Profil-

- ing of Wireless Embedded Systems,” in *2013 ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2013, pp. 153–165.
- [23] M. Schuss, C. A. Boano, and K. Römer, “Moving Beyond Competitions: Extending D-Cube to Seamlessly Benchmark Low-Power Wireless Systems,” in *2018 IEEE Workshop on Benchmarking Cyber-Physical Networks and Systems (CPSBench)*, 2018, pp. 30–35.
- [24] R. Jacob, C. A. Boano, U. Raza, M. Zimmerling, and L. Thiele, “Towards a Methodology for Experimental Evaluation in Low-Power Wireless Networking,” in *Proceedings of the 2nd International Workshop on Benchmarking Cyber-Physical Systems and Internet of Things (CPS-IoTBench)*, 2019, pp. 18–23.
- [25] M. Zimmerling, L. Mottola, and S. Santini, “Synchronous Transmissions in Low-Power Wireless: A Survey of Communication Protocols and Network Services,” *ACM Computing Surveys*, vol. 53, no. 6, dec 2020.
- [26] K. Leentvaar and J. Flint, “The Capture Effect in FM Receivers,” *IEEE Transactions on Communications*, vol. 24, no. 5, pp. 531–539, 1976.
- [27] V. S. Rao, M. Koppal, R. V. Prasad, T. Prabhakar, C. Sarkar, and I. Niemegeers, “Murphy loves CI: Unfolding and Improving Constructive Interference in WSNs,” in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, 2016, pp. 1–9.
- [28] Y. Wang, Y. He, X. Mao, Y. Liu, and X.-y. Li, “Exploiting Constructive Interference for Scalable Flooding in Wireless Networks,” *IEEE/ACM Transactions on Networking*, vol. 21, no. 6, pp. 1880–1889, 2013.
- [29] F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh, “Efficient Network Flooding and Time Synchronization with Glossy,” in *Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks*, 2011, pp. 73–84.
- [30] C. A. Boano, M. Schuß, and K. Römer, “EWSN Dependability Competition: Experiences and Lessons Learned,” *IEEE Internet of Things Newsletter*, Mar. 2017.
- [31] M. Trobinger, D. Vecchia, D. Lobba, T. Istomin, and G. P. Picco, “One Flood to Route Them All: Ultra-fast Convergecast of Concurrent Flows over UWB,” in *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, ser. SenSys ’20. New York, NY, USA: Association for Computing Machinery, 2020, p. 179–191.
- [32] B. A. Nahas, A. Escobar-Molero, J. Klaue, S. Duquennoy, and O. Landsiedel, “BlueFlood: Concurrent Transmissions for Multi-Hop Bluetooth 5 – Modeling and Evaluation,” *ACM Transactions on Internet of Things*, vol. 2, no. 4, 2021.
- [33] M. Bor, J. Vidler, and U. Roedig, “LoRa for the Internet of Things,” in *Proceedings of the 2016 International Conference on Embedded Wireless Systems and Networks*, ser. EWSN ’16. USA: Junction Publishing, 2016, p. 361–366.
- [34] C.-H. Liao, G. Zhu, D. Kuwabara, M. Suzuki, and H. Morikawa, “Multi-Hop LoRa Networks Enabled by Concurrent Transmission,” *IEEE Access*, vol. 5, pp. 21 430–21 446, 2017.
- [35] W. Du, J. C. Liando, H. Zhang, and M. Li, “Pando: Fountain-Enabled Fast Data Dissemination With Constructive Interference,” *IEEE/ACM Transactions on Networking*, vol. 25, no. 2, p. 820–833, apr 2017.
- [36] M. Mohammad and M. C. Chan, “Codecast: Supporting Data Driven In-Network Processing for Low-Power Wireless Sensor Networks,” in *2018 17th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2018, pp. 72–83.
- [37] D. J. MacKay, “Fountain Codes,” *IEE Proceedings-Communications*, vol. 152, no. 6, 2005.
- [38] M. Luby, “LT Codes,” in *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.*, 2002, pp. 271–271.
- [39] T. Ho, M. Medard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, “A Random Linear Network Coding Approach to Multicast,” *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4413–4430, 2006.
- [40] B. Al Nahas, S. Duquennoy, and O. Landsiedel, “Concurrent Transmissions for Multi-Hop Bluetooth 5,” in *Proceedings of the 2019 International Conference on Embedded Wireless Systems and Networks*, ser. EWSN ’19. USA: Junction Publishing, 2019, p. 130–141.
- [41] C. Herrmann, F. Mager, and M. Zimmerling, “Mixer: Efficient Many-to-All Broadcast in Dynamic Wireless Mesh Networks,” in *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys ’18. New York, NY, USA: Association for Computing Machinery, 2018, p. 145–158.
- [42] C. G. Ramirez, A. Dyussenova, A. Sergeyev, and B. Iannucci, “Long-ShoT: Long-Range Synchronization of Time,” in *2019 18th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2019, pp. 289–300.
- [43] J. Heirbaut, “JojoDiff - diff utility for binary files,” [Online] <http://jojo.diff.sourceforge.net/> – Last accessed: 2024-02-08.
- [44] J. Jongboom, “Jojo AlterNative Patch (JANPatch),” [Online] <https://github.com/janjongboom/janpatch> – Last accessed: 2024-02-08.
- [45] Y. Liu, X. Ma, L. Shu, Q. Yang, Y. Zhang, Z. Huo, and Z. Zhou, “Internet of Things for Noise Mapping in Smart Cities: State of the Art and Future Directions,” *IEEE Network*, vol. 34, no. 4, pp. 112–118, 2020.
- [46] C. A. Boano, M. Cattani, and K. Römer, “Impact of Temperature Variations on the Reliability of LoRa: An Experimental Evaluation,” in *Proceedings of the 7th International Conference on Sensor Networks (SENSORNETS)*, 2018, pp. 39–50.
- [47] M. Cattani *et al.*, “An Experimental Evaluation of the Reliability of LoRa Long-Range Low-Power Wireless Communication,” *Journal of Sensor and Actuator Networks (JSAN)*, vol. 6, no. 2, 2017.
- [48] P. Tian, F. Yang, X. Ma, C. A. Boano, X. Tian, Y. Liu, and J. Wei, “Environmental Impact on the Long-Term Connectivity and Link Quality of an Outdoor LoRa Network,” in *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys ’21. New York, NY, USA: Association for Computing Machinery, 2021, p. 565–568.
- [49] A. M. Yousuf, E. M. Rochester, and M. Ghaderi, “A Low-Cost LoRaWAN Testbed for IoT: Implementation and Measurements,” in *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, 2018, pp. 361–366.
- [50] Z. Wang, Z. Xu, B. Dong, W. Xu, and J. Yang, “Dandelion: An Online Testbed for LoRa Development,” in *2019 15th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN)*, 2019, pp. 439–444.
- [51] I. Rodríguez, M. Lauridsen, G. Vasluiuanu, A. N. Poulsen, and P. Mogensen, “The Gigantum Smart City Living Lab: A Multi-Arena LoRa-based Testbed,” in *2018 15th International Symposium on Wireless Communication Systems (ISWCS)*, 2018, pp. 1–6.
- [52] A. van den Bossche, R. Dalcé, and T. Val, “LocURa4IoT—A Testbed Dedicated to Accurate Localization of Wireless Nodes in the IoT,” *IEEE Sensors Journal*, vol. 22, no. 6, pp. 5437–5446, 2022.
- [53] N. S. Senol and A. Rasheed, “A Testbed for LoRa Wireless Communication between IoT Devices,” in *2023 11th International Symposium on Digital Forensics and Security (ISDFS)*, 2023, pp. 1–6.
- [54] S. Sobot, M. Lukic, D. Bortnik, V. Nikic, B. Lima, M. Beko, and D. Vukobratovic, “Two-Tier UAV-based Low Power Wide Area Networks: A Testbed and Experimentation Study,” in *2023 6th Conference on Cloud and Internet of Things (CIoT)*, 2023, pp. 85–90.
- [55] R. Trüb, R. Da Forno, T. Gsell, J. Beutel, and L. Thiele, “Demo Abstract: A Testbed for Long-Range LoRa Communication,” in *2019 18th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2019, pp. 342–343.
- [56] UMBRELLA, “The UMBRELLA Testbed,” 2022. [Online]. Available: <https://www.umbrellaiot.com/what-is-umbrella/umbrella-testbed/>—Last accessed:2024-02-08
- [57] Q. Lone, E. Dublé, F. Rousseau, I. Moerman, S. Giannoulis, and A. Duda, “WiSH-WaT: A Framework for Controllable and Reproducible LoRa Testbeds,” in *2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2018, pp. 1–7.
- [58] J. S. E. A. Sikora, M. Schappacher, and Z. Amjad, “Test and Measurement of LPWAN and Cellular IoT Networks in a Unified Testbed,” in *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, vol. 1, 2019, pp. 1521–1527.
- [59] M. Alipio and M. Bures, “Current Testing and Performance Evaluation Methodologies of LoRa and LoRaWAN in IoT Applications: Classification, Issues, and Future Directives,” *Internet of Things*, vol. 25, p. 101053, 2024.
- [60] M. Baddeley, C. A. Boano, A. Escobar-Molero, Y. Liu, X. Ma, V. Marot, U. Raza, K. Römer, M. Schuss, and A. Stanoev, “Understanding Concurrent Transmissions: The Impact of Carrier Frequency Offset and RF Interference on Physical Layer Performance,” *ACM Transactions on Sensor Networks*, vol. 20, no. 1, 2023.
- [61] E. Soprana, M. Trobinger, D. Vecchia, and G. P. Picco, “Network On or Off? Instant Global Binary Decisions over UWB with Flick,” in *Proceedings of the 22nd International Conference on Information Processing in Sensor Networks*, ser. IPSN ’23. New York, NY, USA: Association for Computing Machinery, 2023, p. 261–273.
- [62] P. Tian, C. A. Boano, X. Ma, and J. Wei, “LoRaHop: Multihop Support for LoRaWAN Uplink and Downlink Messaging,” *IEEE Internet of Things Journal*, vol. 10, no. 17, pp. 15 376–15 392, 2023.