SC-Canopy: A Tiered Network Architecture for Indoor Robot Swarms

Mengyao Liu*
mengyao.liu@cs.kuleuven.be
DistriNet, KU Leuven, Belgium

Luiz Sampaio*
luiz.sampaio@analog.com
Analog Devices International, Ireland

Jonathan Oostvogels
jonathan.oostvogels@kuleuven.be
DistriNet, KU Leuven, Belgium

Kate O'Riordan
Kate.ORiordan@analog.com
Analog Devices International, Ireland

Sam Michiels
sam.michiels@kuleuven.be
DistriNet, KU Leuven, Belgium

Thomas Watteyne
thomas.watteyne@inria.fr
Inria, AIO team, Paris, France

Danny Hughes
danny.hughes@kuleuven.be
KU Leuven, Belgium

Abstract

Swarm Robotics uses collections of simple robots, which collaborate to accomplish tasks faster, and at a lower cost than more complex robot individuals. A major segment of the Swarm Robotics market will operate indoors; patrolling secure facilities, sorting packages in a warehouse or cleaning commercial buildings. As the swarm focuses on collaboration, effective communication is essential. The cost of installing power and network cables thus poses a significant barrier to entry for facilities lacking wireless networking. We tackle this problem by introducing a 'peel and stick' backhaul for swarm robot telemetry that operates for several years on a D-cell battery. Our solution is tiered, using a time-synchronized mesh network as its backhaul and near-field communication between the robots and the mesh, with μW-scale listening power. Our evaluation demonstrates that this architecture achieves over 90% reliability, low power consumption and long battery life.

CCS Concepts

• **Networks → Physical links**; **Wireless access points, base stations and infrastructure**; **Link-layer protocols**.

Keywords

Swarm Robotics, Internet of Things, Wireless Networking

1 Introduction

Swarm robotics uses collections of simple robots that collaborate and coordinate to accomplish a task; often with greater efficacy than a more complex and capable single robot. Examples include warehouse fulfillment [11] and security patrolling [7] among others. Communication is essential for the swarm to enable collaboration, yet the swarm is often deployed to 'greenfield' spaces that lack reliable wireless networking. In this paper, we explore how *wireless mesh networks* can efficiently cover complex indoor spaces to provide reliable network coverage while eliminating power cables and guaranteeing multi-year battery life. Achieving this 'peel and stick' vision for mobile robotics requires us to address a fundamental contradiction: Low power operation demands Radio Duty Cycling (RDC), yet nodes must be awake and listening at all times to receive messages from robots as they unpredictably move across the mesh.

Our approach uses different networking approaches for robot-to-canopy communication, which requires a low power always-listening network, and canopy-to-gateway communication, which requires a reliable multi-hop backhaul. Specifically, we build our approach on a combination of Time Synchronized Channel Hopping (TSCH) mesh network using SmartMesh IP [13], which provides a ceiling-mounted *'canopy'* and Near Field Communication (NFC) in the VHF frequency range (40 MHz) to communicate from mobile robots to the canopy using the Capacitive and Inductive Network (CaIN) [4], which enables μW scale always-on listening.

The scientific contributions of this paper are twofold: *(i.)* We are the first to apply NFC communication for mobile swarm robotics. *(ii.)* We provide a reference design for the integration of NFC communication with low-power mesh networks. This is supported by two engineering contributions: *(iii)* We develop an integrated packet-based radio for CaIN using the existing front end [4]. *(iv)* Our evaluation demonstrates good reliability, and low power consumption.

The remainder of this paper is structured as follows. Section 2 discusses related work. Section 3 describes the design and implementation of our approach. Section 4 evaluates our solution. Finally, Section 5 concludes and discusses directions for future work.

2 Related Work

In this section, we discuss archetypal time-synchronized and asynchronous mesh protocols in Sections 2.1 and 2.2, respectively. We then discuss how technologies inspired by Wake-Up Radios can be used to support mobile devices in Section 2.3. Finally, we enumerate design requirements for SC-Canopy in Section 2.4.

2.1 Time-Synchronized Mesh Networking

The IETF 6TiSCH (IPv6 over the TSCH mode of IEEE 802.15.4e) protocol suite is rooted in the Time Slotted Channel Hopping (TSCH) mode of IEEE 802.15.4. TiSCH integrates a synchronized Medium Access Control (MAC) layer with an IPv6-ready upper stack to deliver industrial-grade performance in constrained environments [12]. TiSCH employs a combination of time- and frequency-division multiplexing, which ensures high reliability and resilience

---

[*] Both authors contributed equally to this work.

to interference and multipath fading, important features for mission-critical industrial applications. The tight synchronization and low-duty-cycle operation of TSCH enables ultra-low power consumption, extending the operational lifetime of battery-powered nodes, while the scheduling function controls how the nodes are connected, with support for multi-hop mesh topologies, allowing the network to adapt to varying traffic demands.

One commercial example of the TiSCH technology is Analog Devices' SmartMesh product lines, which exhibit over 99.999% wire-like end-to-end reliability, a decade of battery lifetime, and certified security [13]. SmartMesh IP has been deployed in over 30,000 networks across more than 120 countries with deployments spanning a range of challenging environments and industries. Despite its strengths, TiSCH exhibits limitations in highly mobile environments due to its reliance on stable parent-child relationships. This makes it less suited for mobile scenarios, which adversely affect packet loss, latency, and energy consumption due to resynchronization and reconfiguration overheads. While enhancements such as adaptive synchronization [3] and mobility-aware scheduling [1] have been explored, TiSCH's current architecture remains optimized for quasi-static deployments.

2.2 Non-Synchronized Mesh Networking

Without time-synchronization, conventional radios use significant power when listening for incoming transmissions and must therefore be duty cycled to maintain an acceptable power profile while being responsive to incoming messages.

Preamble sampling and low-power probing techniques such as Berkeley MAC [10] require one of a pair of devices announces their intent to communicate by transmitting a bit pattern of length L in time, while the other device samples the channel at a rate of *R*, where R≤L. Preamble sampling enables the receiver to be arbitrarily duty cycled at the expense of additional power consumption and latency for the sender. Longer preambles will result in lower power consumption for receivers at the expense of latency. Various routing protocols may be layered on top of this family of MAC protocol. For quasi-static networks, RPL [14] is a leading example, while AODV [8] provides support for mobile networks. Unfortunately, this family of uncoordinated approaches suffer from excessive latency and collisions [13] and cannot implement coherent channel hopping strategies to mitigate sporadic interference and multi-path fading.

2.3 Wake-up Radios

A Wake-up Radio (WuR) [9] is specialized for ultra low power listening, at the expense of throughput and transmitter power. Due to their poor receiver performance, they are generally paired with a conventional radio which takes over communication once the wake-up signal is received. For instance, Bdiri *et al.* [2] introduce a 868 MHz receiver, which achieves a range of 2.5 m, data rate of 1.2 kbps and a power consumption of 690 nW. Similarly, Marinkovic *et al.* [6] introduce a 433 MHz WuR, which achieves a range of 10 m, a data rate of 5.5 kbps and a power consumption of 270 nW. Magno *et al.* [5] introduce an 868 MHz WuR which achieves a range of up to 50 m, a data rate of 10 kbps and receiver power of 1.2 μW. WuR solve an important element of the mobility problem as they enable a receiver to continually listen for messages and therefore remain alive and responsive as nodes unpredictably move in and out of their range. However, their limited performance profile is inadequate to support mobile robotics scenarios.

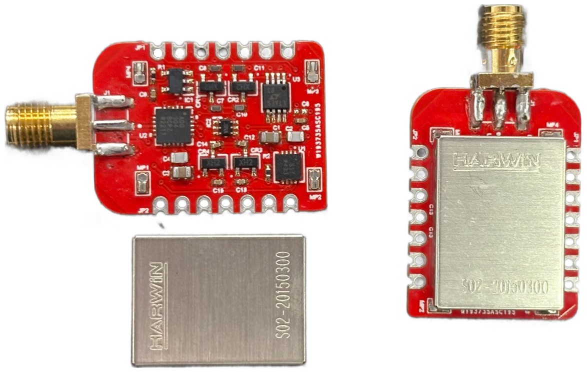

Figure 1: The CaIN transceiver as a packet-based UART radio

In this work, select an ultra low power receiver that can be paired with a time-synchronized backhaul in order to provide the 'best of both worlds', with a *time synchronized canopy* that delivers high reliability, determinism and ultra low-power operation that can provide always-on reception of data transmitted by *mobile tags*, thereby addressing a key problem for time-synchronized mesh networking. The Capacitive and Inductive Network (CaIN) [4] applies a WuR-like design to realize a general purpose radio with an interesting performance profile, including sub-μW receiver power consumption, 20 mW transceiver power, 4.92 kbps application-level throughput, and 200 μs wake-up on message receipt. We extend a swarm of mobile robots with CaIN transceivers *(*mobile tags) which are used to communicate with a canopy of SmartMesh IP nodes likewise extended with CaIN transceivers.

2.4 Requirements

We identify the following requirements for the design of SC-Canopy, which are poorly served by the related work discussed above:

1. Low canopy power consumption, with at least 10 years of battery life when using a D-format LiSO2 battery, enabling peel-and-stick operation with no power or network cables.
2. An adequate range to support a typical indoor ceiling of up to 5 meters.
3. End-to-end reliability of 90% for mobile robot telemetry.

In the following section, we elaborate on the design of SC-Canopy, a system which fulfills these requirements.

3 Design and Implementation

This section describes the design and implementation of *SC-Canopy*. Section 3.1 describes the design of the mobile CaIN transceivers ('tags'). Section 3.2 describes the design of the TSCH mesh network backhaul. Finally, Section 3.3 provides key implementation details.

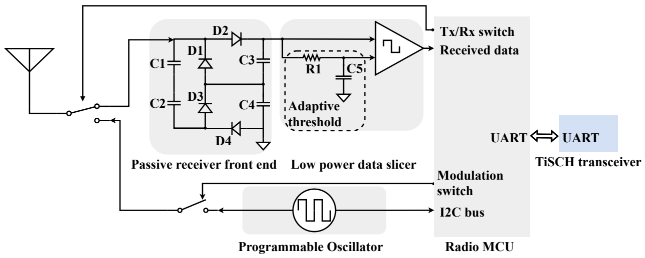

Figure 2: High-level block diagram of the CaIN packet radio

3.1 CaIN tags

Fig. 1 shows the CaIN transceiver used in this paper, from hereon referred to as a ‘tag’. The hardware design of the tag is described in Section 3.1.1, the software design in Section 3.1.2.

3.1.1 Hardware. Fig. 2 shows a high-level block diagram of the CaIN tag, the element of which are detailed below.

1. Passive receiver front end: The CaIN receiver front-end is based on a Greinacher voltage multiplier, which operates as a capacitive charge pump circuit, rectifying received signals and increases their voltage to a usable level. This is a common design in wake-up radios [9].

2. Low power data slicer: The low power data slicer comprises a sub-μW comparator that thresholds a received analog (on-off keyed) signal and thereby converts it to a sequence of digital symbols. An RC low pass filter is applied to one of the comparator inputs to support dynamic thresholding of the input signal, thereby mitigating background noise.

3. Programmable Oscillator: A programmable oscillator is employed to generate a Very High Frequency (VHF) carrier signal within the range of 12 MHz to 40 MHz, thereby enabling access to three unlicensed frequency bands (27, 35 and 40 MHz) [4]. This carrier signal is on-off keyed through an RF switch before transmission, without amplification: at operating ranges below 15 m, unamplified signals can be received reliably due to the near-field coupling between CaIN transceivers [4].

4. Radio micro-controller: A radio micro-controller implements modulation and demodulation in software, and interfaces with the TSCH transceiver using a command/response UART API to accept/deliver data packets to be sent/received. The former micro-controller also configures the frequency of the oscillator through an I2C interface. We selected the Ambiq Micro Apollo 3[1] due to its low power consumption of 6 μA/MHz.

3.1.2 Software. Fig. 3 illustrates the firmware running on CaIN’s radio microcontroller, which implements a simple finite state machine cycling, as described below.

1. Idle Listening: The radio controller remains in an ultra low-power deep-sleep mode. A rising edge on the output of the comparator wakes the device and triggers a transition to the *Reception* state.

2. Preamble Detection: Upon arrival in this state, the radio MCU samples the data output for a period of time corresponding to 4 cycles of the on-off keyed sequence that makes up a CaIN preamble [4]. If the preamble test fails, the wake-up is considered a *false positive* and the radio MCU returns to the Idle Listening State to save energy. If the expected edge pattern is present, the node continues listening for the full frame

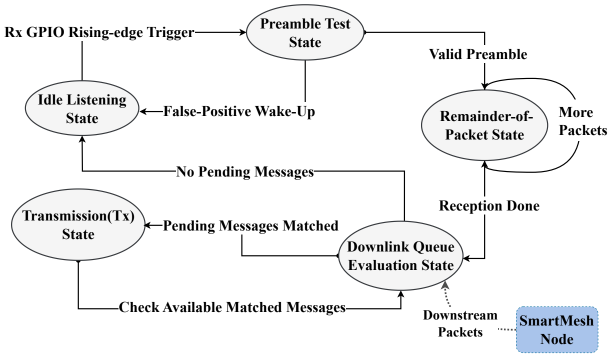

Figure 3: Software state diagram of the CaIN radio firmware.

3. Remainder-of-Packet: Once the expected edge pattern is confirmed, the node keeps listening to capture the rest of the frame. When reception completes, the radio MCU verifies the checksum and if a match occurs forwards the packet upstream over the UART connection of the multi-hop SmartMesh-IP network. In the case that subsequent packets are available, they are processed immediately. All packets are addressed to the SmartMesh-IP gateway, which hosts a simple application to log CaIN packets.

4. Downlink Queue Evaluation: Downstream packets received over UART from SmartMesh are enqueued for later transmission. When an upstream message is received from a robot with a given address, two outcomes are possible: (i) *Pending messages matched* – upon finding a queued frame whose destination address matches the source address, triggering Transmission State; (ii) *No pending messages* – the queue is empty or the address does not match, and the system drops directly back to Idle.

5. Transmission (Tx): In this state, downstream packets are transmitted consecutively: after each one the firmware returns to State 4 to look for further address matches. If the queue is empty, the system resumes idle listening.

3.2 TSCH Backhaul

We use SmartMesh IP as the TSCH backhaul. SmartMesh IP is a product line of Analog Devices. A SmartMesh network offers over 99.999% end-to-end reliability, over a decade of battery lifetime on a pair of AA batteries, and certificated security.

Our evaluation uses the LTC5800-IPM hardware module, based around an ARM Cortex M3 processor and integrated 802.15.4e transciver. This module runs the unmodified SmartMesh-IP software stack which implements all time-synchronisation, medium access control and routing functionality. The LTC5800-IPM is configured to operate in slave mode, offering a UART API to the CaIN microcontroller. All upstream packets received via CaIN are immediately dispatched to the SmartMesh network, while downstream packets received via SmartMesh are likewise immediately dispatched downstream to all CaIN *tags* within range.

---

[1] https://ambiq.com/apollo3/

3.3 Implementation

This section describes a prototypical implementation of CaIN, detailing the Application MCU, firmware stack, and a costed bill-of-materials for a 10,000-unit production run.

CaIN radio MCU. CaIN employs Ambiq's ultra-low-power Apollo3 Blue SoC (AMA3B1KK-KBR[2]), an Arm Cortex-M4F running at 48 MHz (96 MHz TurboSPOT) with 2 MB flash and 384 kB SRAM. The MCU draws roughly 6 µA / MHz in active mode and just 1 µA in deep-sleep with the RTC enabled. Apart from the single I^2C bus that programs the LTC6904 carrier oscillator, only three GPIO lines are required: one selects the transceiver mode, one drives the TX-toggle, and one captures RX-data, leaving most on-chip resources free for future extensions.

Firmware SDK. Firmware is built with *AmbiqSuite–SDK v3.2.0* and the open-source `arm-none-eabi-gcc` toolchain (GCC 11) using standard Makefiles. A bare-metal programming approach is adopted: the CMSIS-compatible HAL and board-support files are leveraged solely for clock, GPIO, DMA, and power management functions, while the application layer executes without an RTOS or BLE stack. This minimalist approach minimises energy consumption and provides predictable, sub-µs interrupt timing by eliminating scheduler overhead.

Bill of Materials. Figure 1 shows the current CaIN transceiver tag, assembled on a 14-pin plug-in header that brings out the SWD flash/debug interface and separate I^2C and UART headers connector for testing CaIN radio with alternative application MCUs. Table 1 lists the hardware bill of materials and the per-unit cost at a production volume of 10,000 boards. The antenna is omitted as it is expected to be customised for each deployment.

Table 1: Bill of materials per unit at a production volume of 10,000 CaIN radios.

Component	Vendor	Unit $ / 10k
Apollo3 Blue SoC (AMA3B1KK)	Ambiq	2.70
SMS7630-005LF Schottky diode	Skyworks	0.57
LTC6904 programmable oscillator	ADI	4.10
TLV7031 analog comparator	TI	0.17
ADG918 SPDT RF switch	ADI	2.46
ADG902 SPST RF switch	ADI	1.49
Passives & ESD devices	—	0.25
Total		**11.74**

4 Evaluation

This section evaluates the performance profile of SC-Canopy in terms of reliability and power in Sections 4.1 to Sections 4.4.

4.1 Static Power Analysis of CaIN

Table 2 summarises the average current consumption of the CaIN canopy node in different operating states. All measurements were obtained by sampling current at 1 MHz using a Joulescope JS220[3].

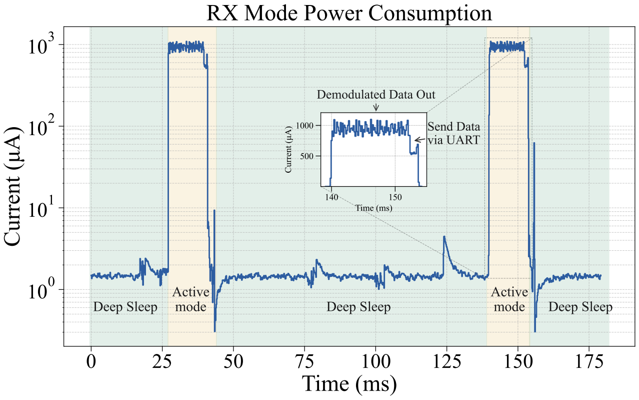

Figure 4: Power profile of the CaIN tag in RX mode. The inset highlights demodulation and UART communication.

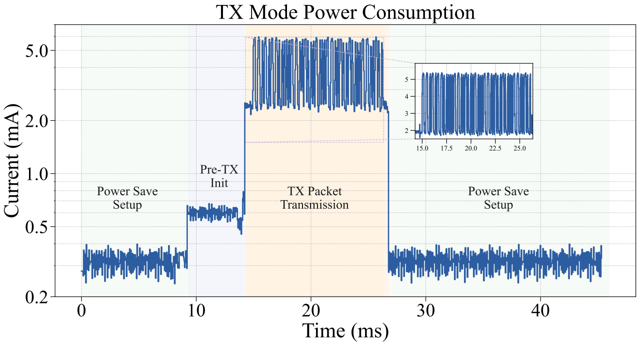

Figure 5: Power profile of the CaIN tag node during Tx mode.

Critically, in *Idle Listening* mode, the MCU remains in deep sleep while the radio module stays active, supporting always-on, low-latency signal detection at a total power budget of 7.1 μW. This enables a rapid response to incoming transmissions compared to time-synchronised or preamble sampling approaches. Active receive and transmit power are comparable to state-of-the-art 802.15.4 and BLE radios such as the LTC5800[4] and the nRF52840[5].

Figures 4 and 5 provide a detailed power timeline for packet reception and transmission respectively. Nodes remain in deep sleep mode whenever they are neither sending nor receiving. Packet reception takes 12.19 ms and consumes 39.5μJ, while transmitting a packet takes 12 ms and 152.6 μJ if the radio is already awake (i.e. the packet is one of a sequence) and 16.88 ms and 157.14 μJ if the radio is in deep sleep mode due to oscillator configuration and settling time. It should be noted that the energy efficiency per bit of CaIN is significantly lower than either BLE or 802.15.4 to its lower data rate. This approach is hence better suited to applications with sporadic traffic where idle listening dominates energy consumption.

---

[2] Apollo3 Blue SoC datasheet
[3] https://www.joulescope.com
[4] https://www.analog.com/media/en/technical-documentation/data-sheets/5800ipmfa.pdf
[5] https://docs.nordicsemi.com/bundle/ps_nrf52840

Table 2: Average power use in different operating states of the CaIN tag.

Mode	Sub-State	Description	Time (ms)	Avg. Power
Rx (Receive)	Idle Listening	MCU in deep sleep; radio module remains active	NA	MCU: 6.02 μW Radio (comparator): 1.08 μW
	Packet Processing	MCU awake, processing received packet	12.19 ms	3.24 mW
	UART Forwarding	Packet forwarded to TSCH transceiver via UART	0.52 ms	1.79 mW
Tx (Transmit)	Power Save Setup	Restore clock, configure transmission GPIO	NA	1.06 mW
	Pre-Tx Init	Tx packet preparation and oscillator warm-up	4.88 ms	0.93 mW
	Transmission	OOK toggling for data transmission	12.00 ms	MCU: 1.66 mW Oscillator: 11.06 mW

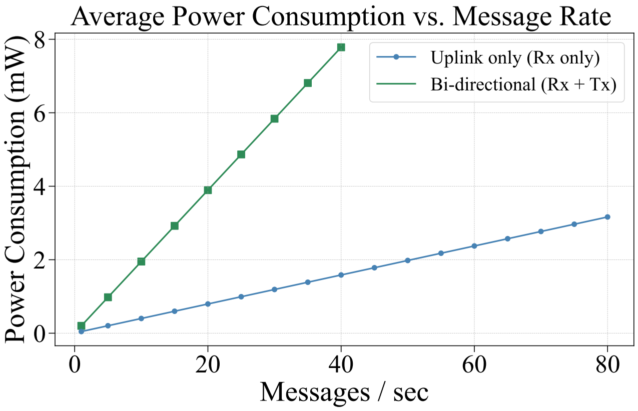

Figure 6: Power consumption vs. message rate (per second) for uplink-only (Rx-only) and bi-directional (Rx+Tx).

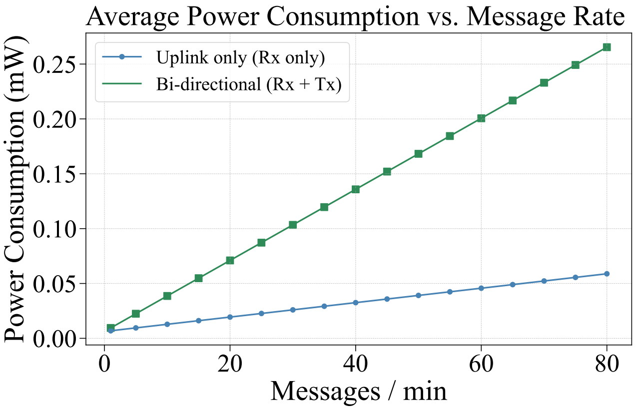

Figure 7: Power consumption vs. message rate (per minute) for uplink-only (Rx-only) and bi-directional (Rx+Tx).

4.2 Dynamic Power Analysis of CaIN

Figures 7 and 6 analyse the power consumption of a two hypothetical protocols with increasing message rates: (i.) an uplink-only protocol wherein mobile robots forward unacknowledged telemetry to the canopy and (ii.) a bi-directional protocol wherein robots forward telemetry upstream to the canopy and receive a single downstream control packet in return. As can be seen from the Figures, CaIN's power consumption is low and scales linearly with traffic load. Bi-directional protocols are much more power hungry as CAIN's primary energy cost lies in message transmission.

4.3 Power Analysis of the SmartMesh IP Canopy

The power profile of SmartMesh IP is modeled using the SmartMesh IP power and performance simulator[6]. We configured the simulator for a 5-hop network of 50 nodes with 10 nodes per hop. Base bandwidth was configured to match the CaIN message rate from 1 to 30 messages per minute. All simulations were performed with 8-byte packets and at 3.3 V. As can be seen from Table 4.3, power consumption is reasonable in all cases, with CaIN delivering always mobile connectivity with a worst-case overhead of 25.9 μW or 4.5% for uplink only and 103.4 μW or 18.1% for bidirectional communication. These results could be improved at the expense of latency by aggregating 8 B CaIN packets prior to transmission in SmartMesh packets of up to 80 B.

Table 3: Power Consumption of SmartMesh IP Canopy

Message Period (s)	Avg. SMIP Power (μW)	CaIN RX Power (μW)	CaIN TX/RX Power (μW)
2	442.2	25.9	103.4
5	219.8	14.1	45.1
15	130.7	8.8	19,4
30	114.2	7.5	12.9
60	109.9	6.9	9.4

Considering battery life, with bidirectional communication, this would equate to approximately 7.5 years of operation using a representative 19 Ah D-cell format LISO2 primary cell[7] when sending messages every 2s and over 100 years when communicating at a rate of one message per minute, which is likely to exceed the useful life of the battery. This supports achieving our target lifetime of 10 years for moderate communication rates.

---

[6] https://www.analog.com/en/resources/technical-articles/smartmesh-power-and-performance-estimator.html

[7] https://www.farnell.com/datasheets/1445890.pdf

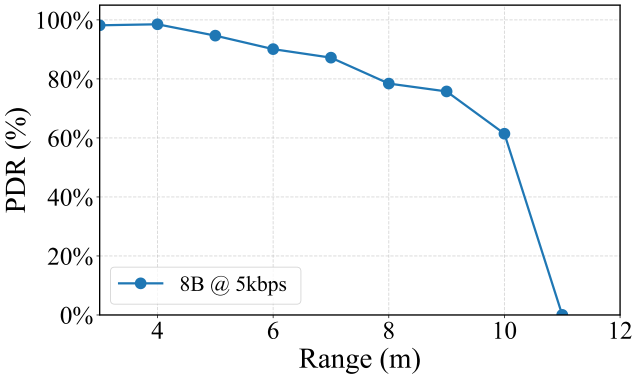

Figure 8: Packet Delivery Ratio (PDR) for 8-byte packets over distances ranging from 1 to 11 meters

4.4 Reliability

Figure 8 shows the Packet Delivery Ratio (PDR) of CaIN for 8-byte packets transmitted at 5 kbps. The PDR remains above 90% up to a distance of 6 m, and then gradually decreasing with range. A sharp decline is observed beyond 10 m, due to near field limitations. This meets our target distance and reliability requirements for a typical indoor ceiling of 5 m. SmartMesh IP reliability always remains over 99.999% in our chosen scenarios; making CaIN the primary driver of packet loss.

5 Conclusions and Future Work

This paper has presented *SC-Canopy*, a tiered network architecture for indoor swarm robotics that combines a ceiling mounted Time Synchronized Channel Hopping (TiSCH) mesh network as a reliable yet low power backhaul and a Capacitive and Inductive Network (CaIN) to gather mobile robot telemetry with a sub-μW power overhead. The resulting network offers seamless robot telemetry while requiring neither power nor network cables. Evaluation shows that SC-Canopy achieves good reliability, low power consumption and long battery life. In sum, these features considerably reduce barriers to entry when deploying indoor swarm robotics applications.

Our future work will focus on three fronts. First, we plan to deploy a much larger SC-Canopy network in the context of the upcoming 1000-node Swarm Robotics testbed at the ADI Catalyst facility. Second, we plan to explore the use of CaIN networking *within* the backhaul network to support low-latency control loops from the robot to the back-end. Third, we aim to miniaturize the CaIN transceiver to support its deployment in a broader range of application scenarios.

Acknowledgments

This work is partially funded by the Research Fund, KU Leuven (C14/24/093; SwarmNet), J. Oostvogels' Research Foundation - Flanders Fellowship (FWO; 1224325N) and is issued within the frame and for the purpose of the Horizon Europe OpenSwarm project[8].

References

[1] Yaarob Al-Nidawi and Andrew H. Kemp. 2015. Mobility Aware Framework for Timeslotted Channel Hopping IEEE 802.15.4e Sensor Networks. *IEEE Sensors Journal* 15, 12 (2015), 7112–7125.

[2] Sadok Bdiri and Faouzi Derbel. 2014. A nanowatt Wake-Up Receiver for industrial production line. In *2014 IEEE 11th International Multi-Conference on Systems, Signals & Devices (SSD14)*. 1–6. doi:10.1109/SSD.2014.6808911

[3] Tengfei Chang, Thomas Watteyne, Kris Pister, and Qin Wang. 2015. Adaptive synchronization in multi-hop TSCH networks. *Computer Networks* 76 (2015), 165–176.

[4] Mengyao Liu, Bingwu Fang, Jonathan Oostvogels, Sam Michiels, Andrei Belogaev, Xinlei Liu, Jeroen Famaey, and Danny Hughes. 2024. CaIN: Low Power and Low Latency VHF Mesh Networking. In *EWSN'24: Proceedings of the 2024 International Conference on Embedded Wireless Systems and Networks*.

[5] Michele Magno, Vana Jelicic, Bruno Srbinovski, Vedran Bilas, Emanuel Popovici, and Luca Benini. 2016. Design, Implementation, and Performance Evaluation of a Flexible Low-Latency Nanowatt Wake-Up Radio Receiver. *IEEE Transactions on Industrial Informatics* 12, 2 (2016), 633–644. doi:10.1109/TII.2016.2524982

[6] Stevan Marinkovic and Emanuel Popovici. 2012. Ultra Low Power Signal Oriented Approach for Wireless Health Monitoring. *Sensors* 12, 6 (2012), 7917–7937. doi:10.3390/s120607917

[7] Alessandro Marino, Lynne Parker, Gianluca Antonelli, and Fabrizio Caccavale. 2009. Behavioral control for multi-robot perimeter patrol: A Finite State Automata approach. In *2009 IEEE International Conference on Robotics and Automation*. 831–836. doi:10.1109/ROBOT.2009.5152710

[8] C.E. Perkins and E.M. Royer. 1999. Ad-hoc on-demand distance vector routing. In *Proceedings WMCSA'99. Second IEEE Workshop on Mobile Computing Systems and Applications*. 90–100.

[9] Rajeev Piyare, Amy L. Murphy, Csaba Kiraly, Pietro Tosato, and Davide Brunelli. 2017. Ultra Low Power Wake-Up Radios: A Hardware and Networking Survey. *IEEE Communications Surveys & Tutorials* 19, 4 (2017), 2117–2157. doi:10.1109/COMST.2017.2728092

[10] Joseph Polastre, Jason Hill, and David Culler. 2004. Versatile low power media access for wireless sensor networks. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems* (Baltimore, MD, USA) *(SenSys '04)*. Association for Computing Machinery, New York, NY, USA, 95–107. doi:10.1145/1031495.1031508

[11] Sumanth Udupa, Ayush Das, and Ajay Victor. 2022. Graph Inspired Geometric Area Allocation for Swarm of Robots in a Warehouse Environment. In *2022 8th International Conference on Control, Automation and Robotics (ICCAR)*. 101–104. doi:10.1109/ICCAR55106.2022.9782664

[12] Xavier Vilajosana, Thomas Watteyne, Tengfei Chang, Mališa Vučinić, Simon Duquennoy, and Pascal Thubert. 2020. IETF 6TiSCH: A Tutorial. *IEEE Communications Surveys & Tutorials* 22, 1 (2020), 595–615.

[13] Thomas Watteyne, Lance Doherty, Jonathan Simon, and Kris Pister. 2013. Technical Overview of SmartMesh IP. In *2013 Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*. IEEE, Taichung, Taiwan, 547–551.

[14] Tim Winter. [n. d.]. RFC 6550: RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks — datatracker.ietf.org. _url-https://datatracker.ietf.org/doc/html/rfc6550. [Accessed 02-05-2024].

---

[8] This project has received funding from the European Union's Horizon Europe Framework under Grant 101093046. Views and opinions expressed are however those of the author(s) only, and the European Commission is not responsible for any use that may be made of the information it contains.