If you cite this paper, please use the ICN reference: P. Kietzmann, J. Alamos, D. Kutscher, T. C. Schmidt, M. Wählisch. Delay-Tolerant ICN and Its Application to LoRa. In *Proc. of ACM ICN*, ACM, 2022.

Delay-Tolerant ICN and Its Application to LoRa

Peter Kietzmann
HAW Hamburg
Germany
peter.kietzmann@haw-hamburg.de

José Alamos
HAW Hamburg
Germany
jose.alamos@haw-hamburg.de

Dirk Kutscher
Hong Kong University of Science and Technology, China
dku@ust.hk

Thomas C. Schmidt
HAW Hamburg
Germany
t.schmidt@haw-hamburg.de

Matthias Wählisch
Freie Universität Berlin
Germany
m.waehlisch@fu-berlin.de

ABSTRACT

Connecting long-range wireless networks to the Internet imposes challenges due to vastly longer round-trip-times (RTTs). In this paper, we present an ICN protocol framework that enables robust and efficient delay-tolerant communication to edge networks. Our approach provides ICN-idiomatic communication between networks with vastly different RTTs. We applied this framework to LoRa, enabling end-to-end consumer-to-LoRa-producer interaction over an ICN-Internet and asynchronous data production in the LoRa edge. Instead of using LoRaWAN, we implemented an IEEE 802.15.4e DSME MAC layer on top of the LoRa PHY and ICN protocol mechanisms in RIOT OS. Executed on off-the-shelf IoT hardware, we provide a comparative evaluation for basic NDN-style ICN [60], RICE [31]-like pulling, and reflexive forwarding [46]. This is the first practical evaluation of ICN over LoRa using a reliable MAC. Our results show that periodic polling in NDN works inefficiently when facing long and differing RTTs. RICE reduces polling overhead and exploits gateway knowledge, without violating ICN principles. Reflexive forwarding reflects sporadic data generation naturally. Combined with a local data push, it operates efficiently and enables lifetimes of >1 year for battery powered LoRa-ICN nodes.

CCS CONCEPTS

• **Networks → Network design principles**; *Wireless access points, base stations and infrastructure*; *Link-layer protocols.*

KEYWORDS

Internet of Things; Information Centric Networks; LPWAN

ACM Reference Format:
Peter Kietzmann, José Alamos, Dirk Kutscher, Thomas C. Schmidt, and Matthias Wählisch. 2022. Delay-Tolerant ICN and Its Application to LoRa. In *9th ACM Conference on Information-Centric Networking (ICN '22), September 19–21, 2022, Osaka, Japan.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3517212.3558081

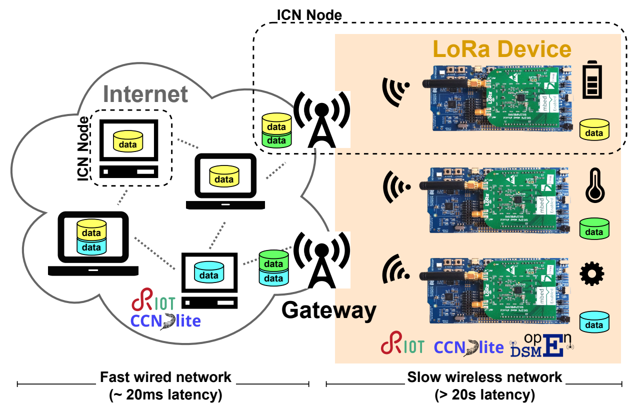


Figure 1: LoRa-ICN network and time domains.

1 INTRODUCTION

LoRaWAN [39] provides a vertically integrated network architecture for connecting LoRa networks and its constrained devices to the Internet that is designed to offload power-constrained wireless LoRa nodes as much possible: gateways relay communication between the wireless link and network servers (often co-located with additional application server infrastructure) that manage the intricate energy-conservation regime of connected LoRa devices.

The energy conservation objectives lead to a MAC layer design that incurs dramatically higher latency and round trip times (RTTs) of several seconds, compared to what connection-oriented Internet transport protocols are typically designed to support. As a result, LoRaWAN supports message-oriented transport through gateways and dedicated network servers only, without a notion of end-to-end communication from the Internet to LoRa nodes. While it is theoretically possible to run bidirectional IP-based communication on top of LoRaWAN [18], the resulting systems inherit latency challenges of LoRaWAN for bidirectional communication that would impact transport layer performance and applicability.

ICN has demonstrated benefits for improving data availability and communication performance in constrained IoT networks [7]. In this paper, we argue that ICN is *also* a suitable network layer for connecting such challenged edge networks to a more regular Internet, by leveraging hop-by-hop transport functions, ICN caching and minimal application-agnostic extensions.

ICN '22, September 19–21, 2022, Osaka, Japan

ACM ISBN 978-1-4503-9257-0/22/09…$15.00
https://doi.org/10.1145/3517212.3558081

Kietzmann *et al.* [28] present a design of an improved, IEEE 802.15.4e DSME [25] based MAC layer for LoRa that supports packet-based communication, specifically ICN-style Interest/Data communication. Yet, RTTs can still be on the order of seconds due to the underlying power saving regime. Leveraging their work, we take an ICN-enabled LoRa subnet as a basis, which is attached via an ICN forwarder on a gateway device. We develop a delay-tolerant ICN communication framework that allows connecting these LoRa sub-networks to a "regular" ICN Internet (Figure 1), with the following design goals: (*i*) supporting IoT sensor data transmission; (*ii*) supporting arbitrary orders of delays, without specific assumptions of typical RTTs on other nodes on the ICN Internet; (*iii*) not requiring application awareness on gateway nodes; (*iv*) utilizing ICN-idiomatic communication to benefit from ICN principles such as accessing named data, Interest/Data semantics, caches, flow balance, etc.

We have developed interactions for IoT communication use cases that leverage bespoke (but application-agnostic) capabilities on gateway-based forwarders and the *reflexive forwarding* extensions for ICN [46]. These cases follow two patterns. First, IoT sensor data retrieval from an Internet-based consumer using Interest/Data interactions; and second, asynchronously "pushing" data from an IoT sensor to an Internet-based consumer with pub/sub semantics. The contributions of this paper are the following:

(1) The design of delay-tolerant ICN-interactions and node behavior for this constrained environment.

(2) A complete implementation of the DSME MAC layer for LoRa [28] and our ICN protocol extensions on RIOT [6], serving common LoRa sensors and RIOT-based gateways https://github.com/inetrg/ACM-ICN-LoRa-ICN-2022.git.

(3) An experiment-based evaluation of the interactions on constrained IoT hardware, connected to an emulated ICN-Internet, and a comparison with vanilla ICN approaches.

The rest of this paper is structured as follows: Section 2 describes essential LoRa and DSME background. Section 3 discusses corresponding challenges that our system design, presented in Section 4, considers. Section 5 introduces our implementation, which is the basis for an experimental evaluation in Section 6. We discuss related work in Section 7 and present our conclusions and future work in Section 8.

2 BACKGROUND

In this section, we describe properties of the LoRa environment and the DSME MAC layer that our work is based on.

2.1 LoRa and LoRaWAN

LoRa defines a chirp spread spectrum modulation which enables a long transmission range (kilometers), low energy consumption (millijoules) at the cost of long on-air times. Duty cycle regulations further limit the effective throughput (bits per second). These features are still attractive for many IoT use cases. We operate on the EU 868 MHz band and configure a spreading factor 7, 125 kHz bandwidth, code rate 4/5, which results in a symbol time of 1.024 ms.

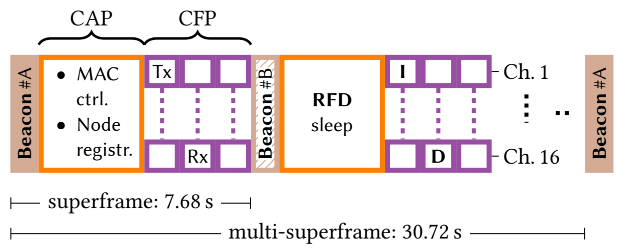


Figure 2: Overview of the DSME multi-superframe structure. Perspective of a coordinator. Exemplary schedule for Interest (I) and data (D).

LoRaWAN [39] is a popular system that operates on top of the LoRa PHY. It defines a vertically integrated, and centralized network architecture to integrate LoRa nodes to the IoT. So-called network- and application servers provide interfacing to the system. A network server interconnects applications and LoRa nodes, via gateways that relay messages from and to the wireless network. The network server organizes MAC schedules centrally, while end devices operate in one of three modes: class A (intended for battery-powered devices) is purely producer-driven, best-effort with very limited support for downlink communication; class C is not suitable for the low-power domain; and class B as a tradeoff between both. LoRaWAN networks are subject to collisions [16, 47] and scalability issues [14, 17]. Class B, albeit rarely deployed, is designed to allow periodic downlink communication at low energy, and exhibits reliability issues [41, 50, 57]. It further reveals long downlink latencies. For example, Elbsir *et al.* [15] measured an average waiting time of 44 s at 26 % delivery ratio in a relaxed class B configuration. All those results motivate re-considering the LoRa MAC system design.

2.2 DSME and LoRa

Motivated by the LoRaWAN deficiencies, we are basing our work on the new DSME-based LoRa MAC design that was introduced by [28]. It has the following key properties that are relevant to this paper:

In DSME (Figure 2), a coordinator emits beacons and initiates a synchronized multi-superframe structure; beacon collisions are inherently resolved for multiple coordinators in reach. Constrained devices (RFD: reduced function device) synchronize to that structure and join the subnet. A superframe is separated into two periods for data transmission: contention-access period (CAP), and contention-free period (CFP). This time division facilitates battery powered nodes to enter low-power mode periodically. CFP slots assign unique and frequency-multiplexed transmission resources between nodes to avoid collisions, and provide a deterministic max. latency. Varying slot assignments enable star-, peer-to-peer-, or clustered tree networks. We focus on star topologies.

For the MAC we configure `macSuperframeOrder`: 3, `macMultisuperframeOrder`: 5, and `macBeaconOrder`: 5. This results in a slotframe structure of four superframes per multi-superframe, a beacon interval and multi-superframe duration of 30.72 s (applying the LoRa symbol time of 1.024 ms from Section 2.1), and provides 28 time slots · 16 frequency channels = 448 exclusive transmission cells. Other slotframe structures trade off subnet size, throughput, energy, and latency; the latter can increase to over 122 s in certain configurations [2].

3 PROBLEM STATEMENT

DSME enables an improved LoRa MAC layer design for reliable bidirectional communication, and it can be configured to provide lower latencies compared to LoRaWAN. As such, it is a much better basis for any packet-based higher-layer network stack, including ICN. Still, due to the energy-conservation objectives and the properties of the underlying LoRa PHY layer, even DSME incurs significant delays for interactive communication, based on its multi-superframe structure. These latencies (30 seconds or more) impose significant challenges to any ICN Interest/Data communication, for example, fetching a sensor value from a LoRa sensor, and will require a delay-tolerant communication system.

Superficially, it seems straight-forward to add delay tolerance to ICN, *e.g.,* by simply adding a face implementation for the DTN (Delay Tolerant Networking) bundle protocol [51] or by implementing a delay-aware forwarding strategy on a forwarder. In reality, NDN [60]- and CCNx [44]-style ICN provides challenges for interconnecting networks with vastly different RTTs, which is mostly due to the dual functions that Interests provide:

(1) Interests and Interest sending rates are central in the transport layer control loop of ICN receiver-driven transport services, *i.e.,* the Interest rate controls the throughput. Interests are used to trigger data transmissions in the first place, and to trigger retransmissions in case no corresponding Data messages have been received within a certain time interval.

(2) Pending Interests are temporary state in forwarders that is needed to implement a symmetric forwarding property in ICN, *i.e.,* to record the downstream face that corresponding Data messages should be forwarded on. A secondary function of pending Interest state it to enable *Interest aggregation* – a feature that would prevent multiple Interests for the same Data object to be forwarded on the same path (when there is current pending Interest for that Data object). *Interest aggregation* effectively means *Interest suppression* for all but the first Interest that has been received by a forwarder in a certain epoch – the Interest lifetime in the Pending Interest Table (PIT) of that forwarder.

For achieving a reliable and decently performing communication service, Interest state on forwarders *has to* expire, otherwise Interest retransmission would *always* be suppressed by on-path forwarders that have pending Interest state (and have not received the corresponding Data object yet). There is a time relationship between the Interest lifetime on forwarders and consumer retransmission timers. For good performance, the Interest lifetime needs to be shorter than the retransmission timer.

To cater to delay-prone networks, one could increase both values, maintaining this property. In a heterogeneous network environment (like the Internet), however, it is impossible to decide on "good values". When connecting a high-RTT edge network to a high-speed and low-RTT Internet, both the Interest lifetime and the Interest retransmission timer would need to be adjusted for the end-to-end path RTT. Alternatively, adaptive suppression mechanisms in forwarders (*e.g.,* implemented in NFD [1]) allow for Interest retransmissions in the presence of matching PIT entries. This does, however, still not solve the problem of guessing suitable timeout values for long and vastly different RTT and adopting these timers on every forwarder. Future research and experiments should further investigate different options.

NDN Interests can provide an optional `InterestLifetime` field that allows a consumer to request more suitable Interest lifetime durations (other than the 4 seconds default). We argue that this is not likely to work well in actual deployments:

(1) Non-edge, high-speed forwarders are not likely to honor non-standard `InterestLifetime` values for individual Interests to avoid the per-packet performance penalty.

(2) In DTN scenarios, RTTs and thus consumer-defined `InterestLifetime` values could be significantly higher than 4 seconds, and a core router may just object to spend memory resources for storing many Interests for a longer time.

(3) In DTN scenarios, the RTT may also change unpredictably, depending on caching, opportunistic contacts, new routing state *etc.* so the `InterestLifetime` and the consumer Interest expiration time would have to be adapted constantly, which could introduce brittleness and inefficiency.

It should be noted that ICN in-network congestion control and specific per-forwarder strategies (for example, delay-tolerant forwarding strategies) do not fundamentally resolve these issues because of the interaction with consumers in the non-challenged network and their different understanding of RTTs and retransmission timers. We argue that, instead of guessing suitable `InterestLifetime` values and hoping for all on-path forwarders to honor the corresponding Interest field, it is better to deal with varying and dramatically higher RTTs (*e.g.,* in DTN scenarios) explicitly, with bespoke ICN protocol mechanisms, without interfering with the ICN network layer Interest lifetime.

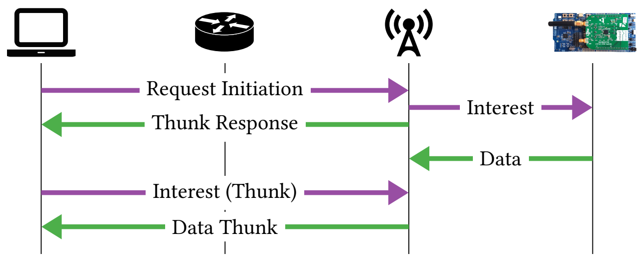
4 SYSTEM OVERVIEW

Figure 1 illustrates our system model: we want to provide ICN delay-tolerant communication to edge networks, such as a LoRa networks so that hosts on the "regular" ICN Internet can communicate (*e.g.,* request data) with hosts in the challenged LoRa edge network, without requiring Internet hosts and forwarders to apply special `InterestLifetime` parameters and retransmission timers.

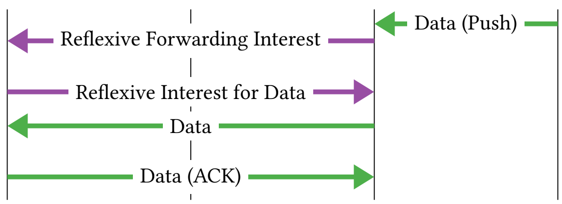
Our work is based on three components: (*i*) a mapping of ICN to DSME, (*ii*) gateway node requirements, and (*iii*) delay-tolerant ICN protocol mechanisms for interconnecting challenged networks (including but not limited to ICN/DSME/LoRa networks) to non-challenged networks – aiming for a seamless integration from an application perspective.

4.1 Mapping of ICN to DSME

DSME provides a contention-access period that is prone to collisions, and a contention-free period (see Section 2) requiring a priori slot negotiation. We exclude node association and dynamic slot allocation from this work, as they are orthogonal to the information-centric and delay-tolerant networking aspects. Evolving [28], we simplify the ICN-DSME mapping and use the CAP only for node



(a) Delay-tolerant Data Retrieval like RICE.



(b) Reflexive Push like *phoning home*.

Figure 3: Delay-Tolerant ICN.

registration (see below), and the CFP for regular network layer traffic since it guarantees exclusive media access. For the CFP traffic, we implement static scheduling. In bidirectional communication, each Interest slot is followed by a data slot. Consequently, presuming data availability, a request is answered withing the same superframe. For unidirectional data push, a single slot is allocated per node.

4.2 Gateway Node Requirements

In our system, a LoRa gateway is an application-agnostic, caching ICN forwarder that connects the narrowband LoRa network to the Internet and follows "regular" ICN behavior (*i.e.,* routing) in the upstream direction. Hence, upstream congestion is uncritical since we consider a broadband network as the default deployment. Downstream congestion on the constrained last hop is handled by the buffering gateway. In addition to regular ICN forwarding and caching, the gateway leverages knowledge about expected delays on the LoRa network for adjusting PIT expiry times and `InterestLifetime` accordingly. This PIT state naturally prevents Interest flooding on the wireless medium, as long as it remains active. Caching, as in other ICN scenarios, offloads (re-) transmissions of Interests and Data messages from the wireless link and the constrained nodes. Moreover, the gateway provides these two additional functions:

Node Registration. LoRa nodes register at the gateway after association, *i.e.,* synchronizing to and joining a network that is advertised by a coordinator. Re-joining a possibly different gateway operates at the order of one (or few) beacon intervals. Nevertheless, it allows for mobile nodes. An overloaded Interest packet by the node indicates its prefix, which establishes a downlink FIB entry on the gateway (see [4]), and the face contains MAC information how to reach that node. Nodes can only serve content under that prefix. On success, the gateway confirms the registration with a data ACK. On a FIB face timeout, *i.e.,* registration expiry, DSME management routines could assist indication (future work).

Local Unsolicited Data. The gateway accepts unsolicited ICN Data messages from registered LoRa nodes and acts as a custodian for these nodes. The corresponding content objects are stored in its CS, and the gateway will respond to corresponding Interest messages from the Internet. Caching strategies manage content placement and timeouts for cache eviction. Although gateways are not constrained in memory, least recently used content items are overwritten in case of overflow.

4.3 Delay-Tolerant ICN Protocols

Delay-tolerant Data Retrieval (Fig. 3a). We want to provide end-to-end ICN communication from an Internet consumer to a LoRa node, *i.e.,* to enable Internet hosts to request arbitrary content objects or to trigger computation in a Remote Method Invocation (RMI) scenario (future work). We leverage the concept of RMI for ICN (RICE [31]) that provides access to static data and dynamic computation results, supporting vastly longer data production/retrieval times. Upon receiving a RICE request initiation Interest, the gateway initiates an Interest message to the LoRa node, as depicted by Figure 3a. A so-called "Thunk Response" contains an indication for the waiting time, leveraging link-knowledge about the DSME configuration in the LoRa network.

Reflexive Push (Fig. 3b). Data generation (*e.g.,* sensor sampling) in the IoT happens sporadically and asynchronously in many cases, which challenges the receiver-driven ("pull") ICN-paradigm [9]. The high LoRa latency further motivates a producer-driven data flow in order to avoid periodic polling. This is consistent with [28] who suggest a unidirectional data push for LoRa-ICN. In this scenario, nodes need to register (as described above) before being authorized to push content to the gateway, using the *Local Unsolicited Data* method. This approach assumes a provisioned name as the *phoning home* destination that could be configured when registering the node at the gateway.

We forward these messages to a node on the Internet by leveraging the *phoning home* use case of the *reflexive forwarding* extension to ICN [46]: the gateway sends an Interest to a configured node on the Internet, which triggers a *reflexive Interest* by that node to retrieve the content object (Figure 3b).

This approach halves the number of resource-intensive wireless transmissions on the last hop, and doubles the number of available DSME slots per multi-superframe. It should be noted that a next-hop signaling does not introduce new security threats, since a network layer can never prevent a malicious neighbor from transmitting unwanted messages (or jamming) on the local link. The slot-based MAC, however, naturally assists prevention of DDoS, triggered by publishing LoRa nodes. A malicious node can simply be muted by the coordinator (*i.e.,* the gateway), de-allocating its CFP slot.

Note: We focus on communication aspects of the protocol mechanisms. Security and corresponding configuration are out of scope for this paper. Hence, we have slightly simplified the protocol operations in our implementation of these schemes (Section 5.2), *e.g.,* we do not use the RICE request parameter retrieval for *Delay-tolerant Data Retrieval.*

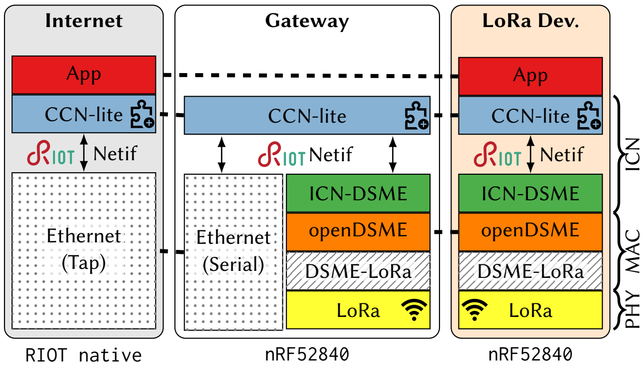


Figure 4: LoRa-ICN stacks on different devices with varying resources and network latencies.

5 IMPLEMENTATION AND DEPLOYMENT

We describe our system implementation in Section 5.1 and the protocol implementation in Section 5.2.

5.1 System Setup

We have implemented this system on actual common off-the-shelf LoRa nodes, and we built our LoRa-ICN gateways on the same constrained hardware, to reduce implementation overhead. In a real-world LoRa network (*cf.* LoRaWAN), however, these gateways are not constrained in energy, memory, or processing power and can serve many low-end nodes simultaneously, through radio concentrators. LoRa devices, gateways, and Internet nodes operate the same ICN stack, to overcome incompatibility issues. In the following, we describe the framework (Figure 4) that we have created for experimentation.

RIOT [6]. We base our implementation on RIOT 2022.04. The networking subsystem (namely GNRC) integrates CCN-lite as an ICN stack, which utilizes the generic network interface layer (RIOT Netif in Figure 4) to send and receive packets. Currently, wired Ethernet and 802.15.4 CSMA/CA wireless interfaces are available. RIOT supports > 230 IoT boards and a *native* port to execute in a Linux process; it utilizes virtual TUN/TAP interfaces for communication. To build CCN-lite based gateways in RIOT that provide both, a fast wired link and a slow long-range radio, we extend the OS integration layer to utilize multiple network interfaces of varying types, behind an ICN face.

CCN-lite [56]. Our integration bases on the latest version, checked out by RIOT 2022.04. CCN-lite provides an ICN forwarder implementation and common data structures: PIT, FIB, and CS. A hop-wise retransmission mechanism re-sends a pending Interest after a pre-configured timeout. Note, received Interest retransmissions will be aggregated when hitting an active PIT entry. PIT state expires after a pre-configured `InterestLifetime` value, as usual. We extend CCN-lite by runtime configuration abilities to adjust the PIT- and retransmission timeout, and the number of retransmissions dynamically. Furthermore, we extend the core forwarder by protocol extensions (🧩) described in Section 4 and the mapping to DSME (ICN-DSME in Figure 4).

openDSME [27]. The open access DSME implementation for 802.15.4 radios was ported to RIOT by Alamos *et al.* [3] who also developed an adaptation layer for LoRa (DSME-LoRa in Figure 4). Their code is publicly available, albeit not on RIOT upstream. We base our work on their implementation and add interfaces to dynamically control MAC parameters (*i.e.,* ACK request, send period) on a per-packet basis, through the RIOT network interface. The southbound interface utilizes the 802.15.4 radio abstraction API of RIOT.

LoRa Device. We deploy the long-range sensor application on common low-power IoT hardware. The Nordic nRF52840 development kit consists of an ARM Cortex-M4 which provides 256 kB RAM, 1 MB flash, and runs at 64 MHz. A SX 1276 LoRa radio shield is attached via pin headers and connects the external radio via SPI. An adjusted transceiver driver implementation exposes the device an 802.15.4 radio, with LoRa specific timing parameters. This facilitates its usage with openDSME. The sensor node is operated as a reduced function device and synchronizes to the DSME multi-superframe, indicated by a coordinator. Afterwards, the node registers its ICN prefix using Interest/Data (see Section 4.2).

Gateway. To reduce implementation overhead, we deploy our gateway on the same hardware as the sensor application. Our gateway acts as a coordinator for LoRa nodes and creates the DSME slotframe structure through the wireless interface. To communicate with a 'fast' infrastructure ICN network in parallel (see forwarder and consumer below), we enable a second network interface; `ethos` is a RIOT specific implementation for Ethernet over serial communication lines. This is required because our experimentation platform lacks Ethernet hardware. Real-world gateways, however, would simply use a gigabit Ethernet link. Our serial device connects to a common Linux based workstation which bridges to a virtual TAP bridge.

Internet (Forwarder and Consumer). Nodes on the Internet are emulated by RIOT-*native* instances to utilize the same ICN stack, and connect to the same virtual TAP bridge as our gateway. We deploy two nodes in a line topology, one forwarder and one consumer. Both run in a *Mininet* [42] emulation to enable short link delays of 20 ms and optional link losses on the virtual wire.

5.2 Protocols for Data Retrieval

We evaluated our system design, comparing its performance to that of regular ICN Interest/Data communication. To that end, we have defined three different data retrieval classes corresponding to Section 4.3:

- *Vanilla ICN Request* for regular Interest/Data interactions initiated from a consumer on the Internet;
- *Delay-tolerant Data Retrieval* using a simplified RICE exchange initiated from a consumer on the Internet;
- *Reflexive Push* using *reflexive forwarding* and the *phoning home* use case initiated from the producer.

Vanilla ICN Request. We assume a regular Interest request from the Internet to the LoRa sensor (Figure 5a). The request faces a non-typical long round trip time at the gateway, conflicting with

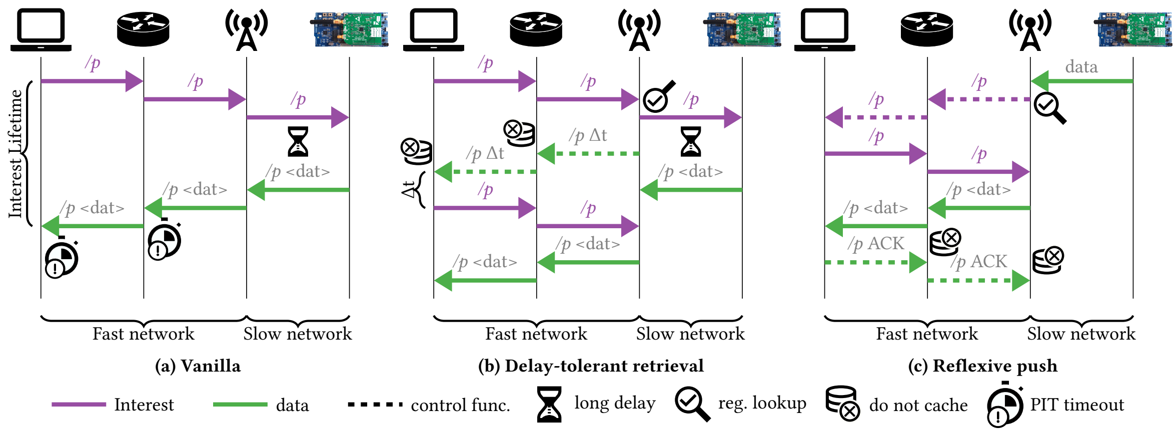


Figure 5: Sequence flows of Interest/Data and ICN extensions between nodes of different time domains. Data flows from LoRa producer to Internet consumer, either initiated by the consumer (5a-5b) or by the producer (5c).

PIT state on forwarders. (*i*) "Regular" forwarders that are not aware of the long delay domain are likely to operate on a fast-network timescale. PIT state that expires before data arrival prevents forwarding on the reverse path. (*ii*) Interest retransmissions are common in ICN, albeit left to transport or application layer implementations. In general, regular ICN-based data retrieval quickly leads to polling and unterminated retransmissions when facing long delays. Two built-in ICN countermeasures are worth discussing: first, `InterestLifetime` dictates the PIT entry expiration time on forwarders. Increasing `InterestLifetime` solves the problem of expired PITs, however, it also requires forwarders to maintain state during the long DSME-LoRa round trip. In addition to occupying PIT memory, this approach affects Interest retransmissions (as a response to timeouts at consumers). Second, common ICN implementations (*e.g.,* RICE, NFD [1]) rely on consumer-based retransmissions (contrasting in-network retransmissions). This *requires* PITs to expire fast, otherwise, a retransmission will be suppressed.

Delay-tolerant Data Retrieval. We have implemented the interaction from Section 4.3 by adding server logic to the link-aware gateway that is triggered by the reception of corresponding Interest messages from consumers in the non-challenged Internet (Figure 5b). The gateway performs three major actions after an incoming Interest: (*i*) It first checks for a registered LoRa node that falls under the requested prefix, in its FIB. (*ii*) On a missing FIB entry, it immediately returns a data NACK. (*iii*) On success, it forwards the Interest as per regular forwarding using the FIB face towards the LoRa node. On forwarding, the gateway replies with a distinct data NACK (we call it WAIT) which contains an estimated data arrival time. A gateway can provide accurate estimates in the future, using its knowledge of the DSME configuration upfront, the internal scheduler state, as well as the current traffic load (queue length). This data packet satisfies the initial Interest, corresponding in-network state, and terminates potentially inappropriate ICN-based retransmissions. The estimated data arrival time enables the consumer application to set an appropriate retry timer, without the need for specific producer knowledge and varying long delays introduced by DSME-LoRa. NACK/WAIT data packets in (*ii*) and (*iii*) must not be cached, though, to prevent serving a subsequent request of the same name from the CS. Finally, after a repeated Interest request, the data item is likely served from the gateway.

Reflexive Push. Our protocol flow (Figure 5c) implements the second interaction from Section 4.3. It suggests two nested Interest/Data exchanges. After successful content placement on the gateway, using *Local Unsolicited Data*, this one indicates data by sending an Interest packet that contains the data name, to the consumer. An additional packet indicator triggers the establishment of a temporary downlink FIB entry on forwarders for that specific name, which points to the incoming face. The consumer can return a *reflexive Interest*, requesting the announced data; it follows the previously established FIB path. Data is served from the gateway cache as usual, satisfying PIT state on the reverse path, and additionally removes the temporary FIB entries. An optional final data ACK terminates the initial Interest request.

6 EVALUATION

We describe experiment configurations in Section 6.1, measurement results for protocol performance in Section 6.2, results from our analysis of communication overhead in Section 6.3, and system overhead of the protocol stacks in Section 6.4.

6.1 Experiment Configuration

We conducted five experiments (comparing our two schemes described in Section 4 with three *Vanilla ICN* variants):

Vanilla (1) Baseline scenario with unchanged ICN and common parameter settings.

Vanilla (2) Delay-aware consumer with extended `InterestLifetime` and retransmission interval.

Vanilla (3) Like **(2)**, additionally forwarders observe the long `InterestLifetime` and set their PIT timer accordingly.

Delay-tolerant retrieval Gateway acts as a special proxy for long-delay producers and returns a distinct re-try instruction on first request.

Reflexive push Producer initiates a transaction by pushing data to gateway CS which triggers a reflexive Interest/Data interaction for retrieving content.

In our experiments, we use unique content names, prefixed with a LoRa node ID and incremental local object counters. Data contains either a random integer value or an ACK, NACK, or WAIT instruction with a time hint. This fixed size scheme leads to a frame size of 31 Bytes for Interest and 36 Bytes for data, which leaves headroom to the maximum frame size of 127 Bytes. Longer packets, however, could be compressed [22] and fragmented [34] in the future. Every content item is requested/indicated once, with an average interval of one minute (60±10 s uniformly distributed). For a fair comparison between consumer- and node initiated traffic, we produce sensor data on the LoRa node after an incoming Interest. Data returns during the subsequent CFP slot within the same superframe (compare Section 4.1). Our measurements include: (*i*) completion time, *i.e.,* the delay between issuing a transaction and data arrival at the consumer; (*ii*) resilience, *i.e.,* the rate of successful transactions; (*iii*) protocol overhead, *i.e.,* the number of transmitted packets per content item. Thereby, we deploy an idealized scenario with 0% – and the case for 5% link loss on the Internet emulation.

Table 1 summarizes our parameter settings. All but the last scenario require the gateway and node to lift the PIT expiration time to the long delay domain. We conservatively chose 60 s which reflects ≈ two times the multi-superframe duration of the MAC (compare Section 2.2). Retransmits on the LoRa hop are disabled since we utilize exclusive CFP resources.

For Vanilla ICN, we distinguish the case with in-network retransmissions (INR) and consumer-based retransmissions (CR), with different PIT timeout behavior. Our **Vanilla (1)** configuration assumes that Internet nods are unaware of the long delay domain. Hence, we set a PIT expiration time of 4 s according to default settings of the common NFD implementation [1] and enable three network layer retransmits, each after 1 s, which reflects the initial round-trip estimation of TCP [48]. In **Vanilla (2)**, a consumer is aware of long producer delays, hence, we set `InterestLifetime` and PIT expiration time to 60 s as well, and adjust the retransmission interval to 15 s. Forwarders do not adopt the long timeout value. In **Vanilla (3)** the forwarder adopts the `InterestLifetime` value of the incoming packet and sets its PIT expiration time accordingly, *i.e.,* to 60 s. This does not change its retransmission behavior in the INR case, however. We present two alternative solutions: **Delay-tolerant retrieval** gets along with 'short' Vanilla (1) parameters and utilizes INR. **Reflexive push** inverts the original ICN semantic and consists of two nested Interest/Data flows that utilize 'short' time parameters analogously.

6.2 Completion Time and Resilience

Figure 6 presents the cumulative distributions of completion times of successful transactions. These values are mainly affected by the multi-superframe duration of the MAC (30.72 s) which dictates the maximum latency of a unidirectional long-range transmission between. Data losses result in infinite completion times, hence, the end value of each graph inherently reflects its success ratio.

Vanilla (1) (Fig 6a). 10–16% of requests are successful and finish in less than the PIT timeout of 4 s. This is the case for Interest that happen to arrive at the gateway short before a DSME transmission slot occurs. Link losses further drop the success rate by 2–6%, but different retransmission pattern do not provide a significant effect.

Table 1: Scenario and parameter overview including four measured nodes. (Abbreviations: INR=In-network retransmission, CR= Consumer retransmission, ✘= not applicable).

Scenario		Cons.		Fwd.		Gw.		Node	
		⏱	↻	⏱	↻	⏱	↻	⏱	↻
Vanilla (1)	INR	4	3:1	4	3:1	60	0:0	60	✘
	CR	4	3:1	4	✘	60	0:0	60	✘
Vanilla (2)	INR	60	3:15	4	3:1	60	0:0	60	✘
	CR	60	3:15	4	✘	60	0:0	60	✘
Vanilla (3)	INR	60	3:15	60	3:1	60	0:0	60	✘
	CR	60	3:15	60	✘	60	0:0	60	✘
Delay-tolerant retrieval	INR[1]	4	3:1	4	3:1	60	0:0	60	✘
Reflexive-push	INR	4	3:1	4	3:1	4	3:1	✘	0:0

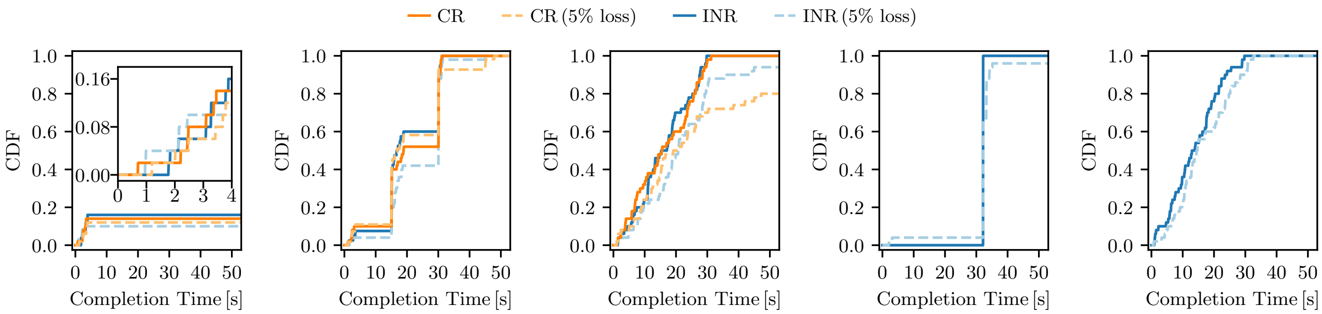
⏱ PIT timeout [s] ↻ Retransmission attempts and timeout [#:s]

[1] Additional retry based on WAIT instruction on first request.

Vanilla (2) (Fig. 6b). Completed transmissions in <4 s resemble properties of the Vanilla (1) case. Steps at 15 s indicate the poll interval of the consumer, which recovers losses from long DSME-LoRa delays. This requires, however, that forwarder PIT state expires fast (here 4 s) to prevent *Interest aggregation.* Losses delay the completion and are compensated faster with INR overall (≈ 32 s), though, CR recovers 20 % more requests on the first retry. Conversely, 10 % of the requests require a third retry with CR, to complete successfully (≈ 45 s). A comparison of CR with and without loss reveals a diverse picture. Here, the lossless case surprisingly satisfies fewer requests after the first retry, which is an effect of randomized experimental requests.

Vanilla (3) (Fig. 6c). Cases without link loss require 32 s at max. (multi-superframe duration) to retrieve all content, which directly reflects the delay distribution of the DSME-LoRa MAC. The long PIT state on both consumer and forwarder allow data forwarding whenever it is ready, reflecting the case for soft-state subscription by a long-lived Interest [11]. Link losses, however, demonstrate the drawback of this approach. CR prevent effective loss recovery while PIT sate is active on the forwarder, and drop the delivery rate to 80 %. INR recover most losses and result in 94 % delivery. This approach only performs well under the assumption that (*i*) every forwarder adopts the long PIT timeout, and (*ii*) content can be retrieved within that time.

Delay-tolerant retrieval (Fig. 6d). Requests finish in almost exactly 32 s in the lossless case, which is the returned WAIT time of the gateway after the first request of a content item. This static worst case value could be reduced with a latency estimator model



(a) Vanilla (1) (b) Vanilla (2) (c) Vanilla (3) (d) Delay-tolerant retrieval (e) Reflexive push

Figure 6: Time to content arrival with long producer delays. Vanilla ICN in varying configurations and our extensions employ in-network retransmissions (INR) or consumer retransmissions (CR), and we vary the link loss.

on the gateway, allowing for targeted completion times. The gateway retrieves content from the node during WAIT, and caches it. A subsequent request of that item is answered from the gateway CS. Additional link losses are mostly recovered by INR and perform similarly to the lossless case, however, two effects are noteworthy: (*i*) ≈ 5 % of the requests finish below 3 s. A loss of the first data packet, which contains a WAIT instruction, triggers an INR which is already satisfied by the gateway. (*ii*) ≈ 8 % of the requests are not satisfied. Our implementation uses a short circular list of future requests to re-issue, which avoids (larger) PIT state over long time. In the loss case, entries stayed longer in the list and got overwritten occasionally, leading to un-requested data. In practice, the list should be provided with timeout values and dimensioned according to traffic load.

Reflexive push (Fig. 6e). Transactions finish in max. 32 s (multi-superframe duration) with 100 % success. Completion times reflect the delay distribution of DSME-LoRa, similarly to the Vanilla (3) scenarios. Herein, the additional round trip of a nested double Interest/Data flow has a negligible overhead when directed towards the fast network. Thereby, losses are smoothly recovered by INR, at minimal time overhead. Contrasting Vanilla (3), this approach works with arbitrary (producer) delays and forgoes the need to adopt long PIT timeout values on Internet nodes.

Findings. Expired PIT state on the reverse path is the prevalent obstacle with vanilla ICN and prevents round tips >4 s, which renders the baseline scenario unusable in this domain. Application-aware consumers overcome long delays, however, the performance heavily depends on the (arbitrary) choice of a poll interval and is susceptible to *varying* delays. Increasing the `InterestLifetime` on the complete forwarding path, instead, is challenging. (*i*) We cannot expect real forwarders to blindly adopt arbitrary PIT timers. (*ii*) Without in-network retransmission in place, long-lived PIT state harms reliability. The Delay-tolerant retrieval case overcomes requirements of long PIT state and blind polling. It thereby relieves Internet nodes and applications from knowledge of the (variable) long time domain. Consumer implementations become more complex, therefore. A reversed transaction flow with Reflexive push facilitates efficient, reliable, and 'timely' transactions.

6.3 Communication Overhead

Figure 7 quantifies the protocol overhead for every node and scenario (cf. Section 6.1) and shows the number of transmitted Interest and Data packets per requested content item as well as the success rate, replicated from Figure 6. In a three hop network, an optimal ICN request-response requires six packets, as indicated by the dashed line. Recall that all scenarios but *Reflexive push* lift the PIT timeout on the gateway and disable network layer retransmissions on the LoRa link, to preserve sparse resources. Consequently, gateways only transmit one Interest towards nodes that respond with one data packet per request.

Vanilla (1). These scenarios reveal notable overheads by futile retransmission, regardless of link loss. Up to two times as many packets are transmitted, compared to the ideal case, with little overall success. With INR, both forwarder and consumer transmit at maximum (4 Interests/content), while CR keeps forwarder overhead low (1 Interest/content). Interests are aggregated as long PIT state persists. Standard retransmit intervals cannot cope with long delays.

Vanilla (2). INR reveal the highest overhead among all scenarios (15 transmissions), sending requests at two timescales. Every consumer Interest is forwarded *and* retransmitted by the forwarder, regardless of the long delay of the producer. In contrast, CR overhead (≈ 9 transmissions) is on par with Vanilla (1) CR but satisfies all requests, without blind forwarding. Hence, PIT timeouts < consumer poll intervals that operate at the prevalent delay domain are a viable option for the conventional ICN paradigm. Short-lived PIT state cannot prevent duplicate data transmission by the gateway, though, when Data faces expired PIT state on a forwarder.

Vanilla (3). INR recovers link losses, while a 'sufficiently' long PIT expiry time prevents consumer-based retransmissions. The CR case (without loss) thus operates with little overhead (≈ 7.5 transmissions) but is not vital due to high sensitivity to link loss.

Delay-tolerant retrieval. Our approach generally increases the required transmissions per content, introducing a second round trip between gateway and consumer. Hence, it performs optimal in the lossless case by transmitting 10 packets: 2xInterest/Data on fast nodes, and 1xInterest/Data on the LoRa link. INR marginally

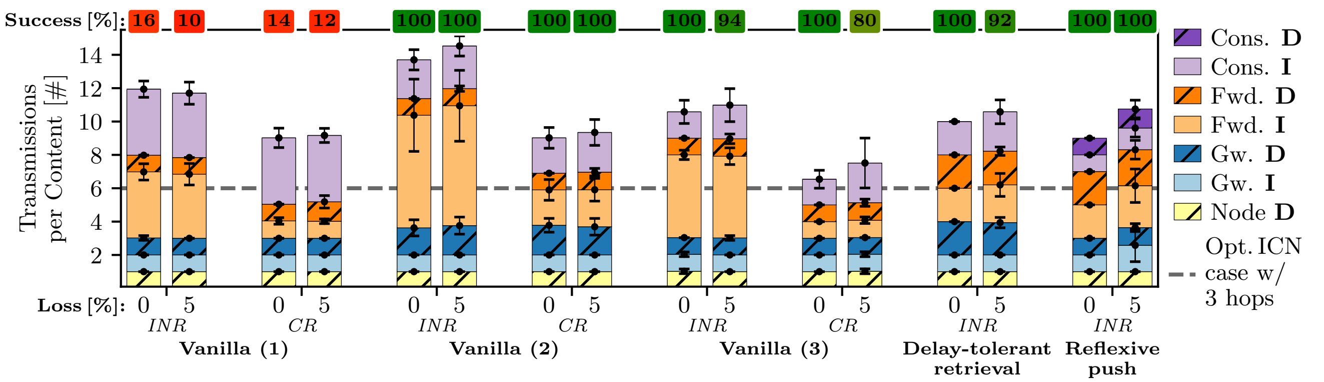


Figure 7: Transmissions per content item (protocol overhead) separated into consumer (Cons.), forwarder (Fwd.), gateway (Gw.), and LoRa node. Vanilla ICN in varying configurations and our extensions employ in-network retransmissions (INR) or consumer retransmissions (CR), and we vary the link loss. Lossless ICN transmission via 3 hops is indicated by the dashed line.

increase the overhead. The total overhead compares to Vanilla (2) with CR, however, it surpasses blind polling.

Reflexive push. Our second approach inverts the flow direction and introduces a second round trip between gateway and consumer as well. In contrast to Delay-tolerant retrieval, however, only a single LoRa transmission is required to place producer content in the gateway cache. This producer oriented optimization results in an optimal number of 9 transmissions per content, reflected by the lossless case. INR increases up to 10.5 transmissions (avg) on loss. Data ACKs by the consumer are optional and terminate the initial Interest of the gateway. Omitting these packets is principally possible to reduce transmissions, however, this conflicts with INR.

Findings. Delay-tolerant retrieval and Reflexive push are robust, operationally efficient, and can tolerate varying delays of the DSME-LoRa MAC. In contrast, vanilla ICN requests suffer from long and unpredictable delays. Naive consumer polling is an inefficient but viable ICN-idiomatic alternative, provided that Interests expire on the forwarding path and polling intervals are set in agreement with practical delays.

6.4 System Overhead

We evaluate the resource overhead of our protocol stack and focus on the battery driven LoRa device, since gateways and Internet nodes are not resource constrained and remain unchallenged by common LoRa traffic.

Energy Consumption. We present the energy consumption per multi-superframe in Table 2, as well as the corresponding nodal lifetimes when operated from an off-the-shelf AA alkaline battery (2800 mAh). Our results are based on extensive measurements performed in [2], which quantify the energy consumption for passive and active periods of the DSME-LoRa superframe structure. Radio operations dominate consumption, *i.e.,* wireless transmission and (idle) reception. To confirm this observation, we also measure the active CPU time throughout our experiments, which is as low as

Table 2: Energy consumption per multi-superframe and lifetime for the protocols under consideration.

Protocol	Energy [mJ]	Lifetime [d]
Vanilla ICN request		
w/o MAC	1247.46	10
w/ MAC	51.42	230
Delay-tolerant data retrieval	51.42	230
Reflexive push	30.83	384

≈ 0.25 % for all protocols on the constrained node, and around 0.3 % on the gateway. The latter increases with growing network sizes.

Vanilla ICN request values include the alternative operation without a MAC (ignoring wireless interference), which strongly motivates the choice of a duty cycling MAC from the energy perspective. Without duty cycling, the lifetime is limited to 10 days. Enabling the MAC reduces the energy consumption by two orders of magnitude, which leads to a lifespan of 230 days in the vanilla ICN request and delay-tolerant data retrieval case, assuming that the gateway shields LoRa devices effectively from retransmits. Reflexive push almost halves the energy consumption due to unidirectional transmission, which further increases the lifetime to more than a year.

Memory Requirements. Our network stack is runtime configurable to operate the three protocols for data retrieval (Section 5.2). Hence, the firmware image is the same for all configurations and requires 143 kB in ROM (text + data segment) and in 19 kB RAM (bss + data segment), almost half of which is occupied by openDSME. The remaining RAM (256 kB on nRF52840) is reserved for dynamic runtime memory allocation (heap). Both openDSME and CCN-lite utilize `malloc`, and we track the combined heap statistics which ranges between 6–8 kB in all experiment runs. Thus, our LoRa-ICN stack can even be deployed on much smaller IoT hardware.

7 RELATED WORK

Advancing LoRa(WAN). To overcome limitations of the centralized LoRaWAN architecture, multi-hop extensions for LoRa [8, 13, 19, 55] have been proposed. These are orthogonal to our work since we focus on single-hop topologies.

Contention-based [35, 45] and scheduled MAC layers [24, 26, 58, 61] for LoRa indicate performance improvements compared to LoRaWAN. Alamos *et al.* [2, 3] re-utilize IEEE 802.15.4e DSME (Deterministic and Synchronous Multi-Channel Extension) [25] to coordinate LoRa radios, with few modifications to the radio configuration. Fixed time-slotted DSME paired with low data rates increases latencies even further, though. In this work, we enable LoRa to run a robust DSME-based MAC layer with latencies that we are able to handle.

RFC 9011 [18] specifies Static Context Header Compression and Fragmentation (SCHC) for IPv6 over LoRaWAN. We agree that compression and fragmentation are crucial, but do not address the latency issues for transport protocols. Also, SCHC does not fix the underlying MAC, which is prone to collisions and depends on network server scheduling.

ICN and the IoT. The IoT benefits from ICN [5, 7, 20, 40, 49, 52, 53, 59]. An important observation in prior work is that IoT scenarios require the adaptation of the MAC layer to prevent unnecessary broadcast and preserve energy resources [29]. Current analyses either base on 802.15.4 CSMA/CA [20], requiring receivers to be always on, or 802.15.4e TSCH [23], allowing for intermittent device sleep.

NDN over LoRa was introduced in [30, 38] which required permanent powering of the nodes, depleting the battery. Unfortunately, latency analyses have not been considered. Recent work [38] shows the need for a MAC protocol due to high collisions even when deploying only few LoRa nodes.

A system design for ICN over DSME-LoRa is proposed in [28]. Based on simulations, the authors find latencies at the order of tens or hundreds of seconds. In this paper, we close the gap and present a solution to handle these high delays and thus enable common, inter-network IoT deployments.

Delay-tolerant ICN. Another ICN application domain that is challenged by long delays are satellite networks. Siris *et al.* [54] find that hop-wise transfer and caching help to increase performance in such networks. They consider an Interest as a long-lived subscription. In contrast, Kumari *et al.* [33] argue that NDN is not viable in satellite scenarios, due to inefficient polling. This is in line with our experimental results. To reduce long delays and needless retransmissions during satellite handovers, the adjustment of the forwarding path is proposed [37]. This solution requires a signal after connecting to a new satellite.

Carofiglio *et al.* [10] exploit link signaling to indicate some kind of loss to trigger a PIT lookup and eventual retransmits, reducing RTTs and redundant retransmits. LoRa lacks such signaling capabilities. We incorporate link awareness in our proposed DSME-LoRa gateway.

Kuai *et al.* [32] propose delay-tolerant NDN forwarding for vehicular networks. Fundamentally, neighbored nodes overhear surrounding traffic and adjust their retransmission procedure based on directional network density. In simulations, the authors assume a relatively high PIT timeout of 50 s. To prevent large PIT tables due to unnecessary long-lasting entries, NACK data packets can include instructions when to retransmit an Interest [12, 43]. Similarly to delay-tolerant networking with NDN [36], the IoT requires a mechanism apart from pure request-response.

Producer-initiated ICN. Burke *et al.* [9] propose push-based sensor data dissemination, accepting names within a distinct namespace on the consumer. Gündogan *et al.* [21] evaluate name indication that triggers a conventional Interest request on the consumer. Król *et al.* [31] introduce a nested 4-way handshake to enable RMI use cases based on ICN principles, and analyze drawbacks from long latencies. This approach is in line with *reflexive forwarding* [46]. We exploit both push and indication concepts in our evaluation.

8 CONCLUSIONS AND FUTURE WORK

Interconnecting networks with vastly different RTTs is challenging for any non-trivial communication system, including ICN. ICN, unlike other frameworks, however, has the unique potential to enable robust communication to nodes in challenged edge networks without requiring application layer relays. In conjunction with an OS-level implementation of ICN (and extensions), DSME, and LoRa, our two protocol mechanisms for Internet consumer-initiated and LoRa producer-initiated communication exhibit high reliability and targeted completion time (compared to Vanilla ICN) when applied to the delay-prone regime. Despite an additional round trip, our evaluations show low overhead of these approaches, by overcoming redundant polling. We leveraged recently proposed gateway behavior (like RICE) and ICN protocol extensions (*reflexive forwarding*), the latter of which serves many other use cases beyond *phoning home* and could be considered a useful standard ICN feature.

This work leads to interesting future research: First, we will integrate an estimator model in the gateway, aiming to reduce the RTT in our *Delay-tolerant retrieval* case. This relieves consumer knowledge, *e.g.,* to estimate domain specific retry timers individually. Second, we will explore security aspects. This includes, but is not limited to, bootstrapping of LoRa nodes and gateways, a secure registration process which requires trust to the gateway, and authentication of a LoRa node before the gateway acts on its behalf. We will further derive a threat model for end-to-end consumer-to-producer security. Third, we will evaluate the scalability and robustness of our ICN protocol framework in more complex topologies (multi gateway, node to node) to demonstrate data sharing benefits. Finally, we want to investigate additional use cases, including Remote Method Invocation on LoRa nodes and multicast-style communication, *e.g.,* for distributing firmware updates to LoRa nodes.

REFERENCES

[1] Alexander Afanasyev, Junxiao Shi, Beichuan Zhang, Lixia Zhang, Ilya Moiseenko, Yingdi Yu, Wentao Shang, Yanbiao Li, Spyridon Mastorakis, Yi Huang, Jerald Paul Abraham, Eric Newberry, Steve DiBenedetto, Chengyu Fan, Christos Papadopoulos, Davide Pesavento, Giulio Grassi, Giovanni Pau, Hang Zhang, Tian Song, Haowei Yuan, Hila Ben Abraham, Patrick Crowley, Syed Obaid Amin, Vince Lehman, Muktadir Chowdhury, , and Lan Wang. 2021. *NFD Developer’s Guide.* Technical Report NDN-0021. NDN. https://named-data.net/publications/techreports/ndn-0021-11-nfd-guide/

[2] Jose Alamos, Peter Kietzmann, Thomas C. Schmidt, and Matthias Wählisch. 2022. DSME-LoRa: Seamless Long Range Communication Between Arbitrary Nodes in the Constrained IoT. *Transactions on Sensor Networks (TOSN)* (2022). https://dl.acm.org/doi/10.1145/3552432

[3] Jose Alamos, Peter Kietzmann, Thomas C. Schmidt, and Matthias Wählisch. 2022. WIP: Exploring DSME MAC for LoRa – A System Integration and First Evaluation. In *23rd IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)* (Belfast, UK). IEEE, Piscataway, NJ, USA. https://doi.org/10.1109/WoWMoM54355.2022.00050

[4] M. Amadeo, C. Campolo, A. Iera, and A. Molinaro. 2014. Named data networking for IoT: An architectural perspective. In *2014 European Conference on Networks and Communications (EuCNC)*. IEEE, Piscataway, NJ, USA, 1–5.

[5] Sobia Arshad, Muhammad Awais Azam, Mubashir Husain Rehmani, and Jonathan Loo. 2019. Recent Advances in Information-Centric Networking-Based Internet of Things (ICN-IoT). *IEEE Internet of Things Journal* 6, 2 (2019), 2128–2158.

[6] Emmanuel Baccelli, Cenk Gündogan, Oliver Hahm, Peter Kietzmann, Martine Lenders, Hauke Petersen, Kaspar Schleiser, Thomas C. Schmidt, and Matthias Wählisch. 2018. RIOT: an Open Source Operating System for Low-end Embedded Devices in the IoT. *IEEE Internet of Things Journal* 5, 6 (December 2018), 4428–4440. http://dx.doi.org/10.1109/JIOT.2018.2815038

[7] Emmanuel Baccelli, Christian Mehlis, Oliver Hahm, Thomas C. Schmidt, and Matthias Wählisch. 2014. Information Centric Networking in the IoT: Experiments with NDN in the Wild. In *Proc. of 1st ACM Conf. on Information-Centric Networking (ICN-2014)* (Paris). ACM, New York, 77–86. http://dx.doi.org/10.1145/2660129.2660144

[8] Maite Bezunartea, Roald Van Glabbeek, An Braeken, Jacques Tiberghien, and Kris Steenhaut. 2019. Towards Energy Efficient LoRa Multihop Networks. In *International Symposium on Local and Metropolitan Area Networks (LANMAN ’19)* (Paris, France). IEEE, Piscataway, NJ, USA, 1–3.

[9] Jeff Burke, Paolo Gasti, Naveen Nathan, and Gene Tsudik. 2014. Secure Sensing over Named Data Networking. In *13th International Symposium on Network Computing and Applications (NCA’14)*. IEEE Computer Society, Washington, DC, USA, 175–180.

[10] Giovanna Carofiglio, Luca Muscariello, Michele Papalini, Natalya Rozhnova, and Xuan Zeng. 2016. Leveraging ICN In-Network Control for Loss Detection and Recovery in Wireless Mobile Networks. In *Proceedings of the 3rd ACM Conference on Information-Centric Networking*. ACM, New York, NY, USA, 50–59.

[11] Antonio Carzaniga, Michele Papalini, and Alexander L. Wolf. 2011. Content-based Publish/Subscribe Networking and Information-centric Networking. In *Proc. of the ACM SIGCOMM WS on Information-centric Networking (ICN ’11)* (Toronto, Ontario, Canada). ACM, New York, NY, USA, 56–61.

[12] Alberto Compagno, Mauro Conti, Cesar Ghali, and Gene Tsudik. 2015. To NACK or Not to NACK? Negative Acknowledgments in Information-Centric Networking. In *24th International Conference on Computer Communication and Networks (ICCCN’15)*. IEEE Press, Piscataway, NJ, USA, 10 pages.

[13] Jeferson Rodrigues Cotrim and Jo ao Henrique Kleinschmidt. 2020. LoRaWAN Mesh Networks: A Review and Classification of Multihop Communication. *Sensors* 20, 15 (2020), 4273.

[14] Houssem Eddin Elbsir, Mohammed Kassab, Sami Bhiri, and Mohamed Hedi Bedoui. 2020. Evaluation of LoRaWAN Class B efficiency for downlink traffic. In *2020 16th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)* (Thessaloniki, Greece). IEEE, Piscataway, NJ, USA, 105–110.

[15] Houssem Eddin Elbsir, Mohammed Kassab, Sami Bhiri, and Mohamed Hedi Bedoui. 2020. Evaluation of LoRaWAN Class B efficiency for downlink traffic. In *16th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob’20)*. IEEE, Piscataway, NJ, USA, 105–110.

[16] Guillaume Ferre. 2017. Collision and packet loss analysis in a LoRaWAN network. In *25th European Signal Processing Conference (EUSIPCO’17)*. IEEE, Piscataway, NJ, USA, 2586–2590.

[17] Joseph Finnegan, Stephen Brown, and Ronan Farrell. 2018. Evaluating the Scalability of LoRaWAN Gateways for Class B Communication in ns-3. In *IEEE Conference on Standards for Communications and Networking (CSCN’18)*. IEEE, Piscataway, NJ, USA, 1–6.

[18] O. Gimenez and I. Petrov. 2021. *Static Context Header Compression and Fragmentation (SCHC) over LoRaWAN.* RFC 9011. IETF.

[19] Nicolas Gonzalez, Adrien Van Den Bossche, and Thierry Val. 2018. Specificities of the LoRa physical layer for the development of new ad hoc MAC layers. In *17th International Conference on Ad Hoc Networks and Wireless (AdHoc-Now’18)* (St Malo, France), Vol. 11104. Springer, Cham, Switzerland, 163–174.

[20] Cenk Gündogan, Peter Kietzmann, Martine Lenders, Hauke Petersen, Thomas C. Schmidt, and Matthias Wählisch. 2018. NDN, CoAP, and MQTT: A Comparative Measurement Study in the IoT. In *Proc. of 5th ACM Conference on Information-Centric Networking (ICN)*. ACM, New York, NY, USA, 159–171. https://doi.org/10.1145/3267955.3267967

[21] Cenk Gündogan, Peter Kietzmann, Thomas C. Schmidt, and Matthias Wählisch. 2018. HoPP: Robust and Resilient Publish-Subscribe for an Information-Centric Internet of Things. In *Proc. of the 43rd IEEE Conference on Local Computer Networks (LCN)* (Chicago, USA). IEEE Press, Piscataway, NJ, USA, 331–334. http://doi.org/10.1109/LCN.2018.8638030

[22] Cenk Gündogan, Peter Kietzmann, Thomas C. Schmidt, and Matthias Wählisch. 2019. ICNLoWPAN – Named-Data Networking in Low Power IoT Networks. In *Proc. of 18th IFIP Networking Conference* (Warsaw, Poland). IEEE Press, Piscataway, NJ, USA, 1–9. http://dx.doi.org/10.23919/IFIPNetworking.2019.8816850

[23] Oliver Hahm, Cedric Adjih, Emmanuel Baccelli, Thomas C. Schmidt, and Matthias Wählisch. 2016. Designing Time Slotted Channel Hopping and Information-Centric Networking for IoT. In *Proc. of 8th IFIP International Conference on New Technologies, Mobility & Security (NTMS)* (Larnaca, Cyprus). IEEE Press, Piscataway, NJ, USA, 5 pages.

[24] Martin Haubro, Charalampos Orfanidis, George Oikonomou, and Xenofon Fafoutis. 2020. TSCH-over-LoRA: long range and reliable IPv6 multi-hop networks for the internet of things. *Internet Technology Letters* 3, 4 (2020), e165.

[25] IEEE 802.15 Working Group. 2016. *IEEE Standard for Low-Rate Wireless Networks.* Technical Report IEEE Std 802.15.4™–2015 (Revision of IEEE Std 802.15.4-2011). IEEE, New York, NY, USA. 1–709 pages.

[26] Thenuka Karunathilake, Asanga Udugama, and Anna Förster. 2021. LoRa-DuCy: Duty Cycling for LoRa-Enabled Internet of Things Devices. In *12th International Conference on Ubiquitous and Future Networks (ICUFN ’21)*. IEEE, Piscataway, NJ, USA, 283–288.

[27] Florian Kauer, Maximilian Köstler, and Volker Turau. 2018. *Reliable Wireless Multi-Hop Networks with Decentralized Slot Management: An Analysis of IEEE 802.15.4 DSME.* Technical Report arXiv:1806.10521. Open Archive: arXiv.org.

[28] Peter Kietzmann, Jose Alamos, Dirk Kutscher, Thomas C. Schmidt, and Matthias Wählisch. 2022. Long-Range ICN for the IoT: Exploring a LoRa System Design. In *Proc. of 21th IFIP Networking Conference* (Catania, Italy). IEEE Press, Piscataway, NJ, USA. https://doi.org/10.23919/IFIPNetworking55013.2022.9829792

[29] Peter Kietzmann, Cenk Gündogan, Thomas C. Schmidt, Oliver Hahm, and Matthias Wählisch. 2017. The Need for a Name to MAC Address Mapping in NDN: Towards Quantifying the Resource Gain. In *Proc. of 4th ACM Conference on Information-Centric Networking (ICN)*. ACM, New York, NY, USA, 36–42. https://dl.acm.org/doi/10.1145/3125719.3125737

[30] Randy King and Rod Sugiyama. 2017. *A New Remote Communications Link to Reduce Residential PV Solar Costs.* Technical Report DE-EE0007592. U.S. Department of Energy Office of Scientific and Technical Information.

[31] Michał Król, Karim Habak, David Oran, Dirk Kutscher, and Ioannis Psaras. 2018. RICE: Remote Method Invocation in ICN. In *Proceedings of the 5th ACM Conference on Information-Centric Networking* (Boston, Massachusetts) *(ICN ’18)*. ACM, New York, NY, USA, 1–11.

[32] Meng Kuai, Xiaoyan Hong, and Qiangyuan Yu. 2019. Delay-tolerant forwarding strategy for named data networking in vehicular environment. *International Journal of Ad Hoc and Ubiquitous Computing* 31, 1 (2019), 12 pages.

[33] Ritika Kumari and R. L. Ujjwal. 2019. Name Data Networking for Interplanetary Internet: An Architectural Perspective. *International Journal of Research in Advent Technology* 7, 5 (2019), 436–441.

[34] Martine Sophie Lenders, Cenk Gündogan, Thomas C. Schmidt, and Matthias Wählisch. 2020. Connecting the Dots: Selective Fragment Recovery in ICNLoWPAN. In *Proc. of 7th ACM Conference on Information-Centric Networking (ICN)* (Montreal, CA). ACM, New York, 70–76. https://doi.org/10.1145/3405656.3418719

[35] Luca Leonardi, Lucia Lo Bello, Filippo Battaglia, and Gaetano Patti. 2020. Comparative Assessment of the LoRaWAN Medium Access Control Protocols for IoT: Does Listen before Talk Perform Better than ALOHA? *Electronics* 9, 4 (2020), 553.

[36] Tianxiang Li, Zhaoning Kong, and Lixia Zhang. 2020. Supporting Delay Tolerant Networking: A Comparative Study of Epidemic Routing and NDN. In *International Conference on Communications Workshops (ICC’20 Workshop)*. IEEE Press, Piscataway, NJ, USA, 6 pages.

[37] Teng Liang, Zhongda Xia, Guoming Tang, Yu Zhang, and Beichuan Zhang. 2021. NDN in Large LEO Satellite Constellations: A Case of Consumer Mobility Support. In *Proceedings of the 8th ACM Conference on Information-Centric Networking*. ACM, New York, NY, USA, 12 pages.

[38] Yaoqing Liu, Laurent Njilla, Anthony Dowling, and Wan Du. 2020. Empowering Named Data Networks for Ad-Hoc Long-Range Communication. In *Wireless and Optical Communications Conference (WOCC’20)*. IEEE, Piscataway, NJ, USA, 1–6.

[39] LoRa Alliance – Technical Committee. 2017. *LoRaWAN 1.1 Specification.* Technical Report. LoRa Alliance. https://lora-alliance.org/sites/default/files/2018-04/lorawantm_specification_-v1.1.pdf

[40] Bertrand Mathieu, Cedric Westphal, and Patrick Truong. 2016. Towards the Usage of CCN for IoT Networks. In *Internet of Things (IoT) in 5G Mobile Technologies*. Springer, Cham, Switzerland, 3–24.

[41] Konstantin Mikhaylov, Juha Petäjäjärvi, and Ari Pouttu. 2018. Effect of Downlink Traffic on Performance of LoRaWAN LPWA Networks: Empirical Study. In *29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'18)*. IEEE, Piscataway, NJ, USA, 6 pages.

[42] Mininet Project Contributors. 2022. Mininet - An Instant Virtual Network on your Laptop (or other PC). http://www.mininet.org/, last accessed 09-06-2022.

[43] Ilya Moiseenko, Lijing Wang, and Lixia Zhang. 2015. Consumer / Producer Communication with Application Level Framing in Named Data Networking. In *Proceedings of the 2nd ACM Conference on Information-Centric Networking* (San Francisco, California, USA) *(ICN '15)*. ACM, New York, NY, USA, 99–108.

[44] M. Mosko, I. Solis, and C. Wood. 2019. *Content-Centric Networking (CCNx) Semantics*. RFC 8569. IETF.

[45] Morgan O'Kennedy, Thomas Niesler, Riaan Wolhuter, and Nathalie Mitton. 2020. Practical evaluation of carrier sensing for a LoRa wildlife monitoring network. In *Proc. of 19th IFIP Networking Conference* (Paris, France). IEEE Press, Piscataway, NJ, USA, 10–18.

[46] David Oran and Dirk Kutscher. 2020. *Reflexive Forwarding for CCNx and NDN Protocols*. Internet-Draft – work in progress 01. IETF.

[47] Charalampos Orfanidis, Laura Marie Feeney, Martin Jacobsson, and Per Gunningberg. 2019. Cross-Technology Clear Channel Assessment for Low-Power Wide Area Networks. In *16th International Conference on Mobile Ad Hoc and Sensor Systems (MASS'19)*. IEEE Computer Society, Washington, DC, USA, 199–207.

[48] V. Paxson, M. Allman, J. Chu, and M. Sargent. 2011. *Computing TCP's Retransmission Timer*. RFC 6298. IETF.

[49] George C. Polyzos and Nikos Fotiou. 2015. Building a reliable Internet of Things using Information-Centric Networking. *Journal of Reliable Intelligent Environments* 1, 1 (2015), 47–58.

[50] Dara Ron, Chan-Jae Lee, Kisong Lee, Hyun-Ho Choi, and Jung-Ryun Lee. 2020. Performance Analysis and Optimization of Downlink Transmission in LoRaWAN Class B Mode. *IEEE Internet of Things Journal* 7, 8 (2020), 7836–7847.

[51] K. Scott and S. Burleigh. 2007. *Bundle Protocol Specification*. RFC 5050. IETF.

[52] Wenato Shang, Alex Afanasyev, and Lixia Zhang. 2016. The Design and Implementation of the NDN Protocol Stack for RIOT-OS. In *Proc. of IEEE GLOBECOM 2016*. IEEE, Washington, DC, USA, 1–6.

[53] Wentao Shang, Adeola Bannis, Teng Liang, Zhehao Wang, Yingdi Yu, Alexander Afanasyev, Jeff Thompson, Jeff Burke, Beichuan Zhang, and Lixia Zhang. 2016. Named Data Networking of Things (Invited Paper). In *Proc. of IEEE International Conf. on Internet-of-Things Design and Implementation (IoTDI)*. IEEE Computer Society, Los Alamitos, CA, USA, 117–128.

[54] Vasilios A. Siris, Christopher N. Ververidis, George C. Polyzos, and Konstantinos P. Liolis. 2012. Information-Centric Networking (ICN) architectures for integration of satellites into the Future Internet. In *First AESS European Conference on Satellite Telecommunications (ESTEL'12)*. IEEE, Piscataway, NJ, USA, 6 pages.

[55] Steffen Thielemans, Maite Bezunartea, and Kris Steenhaut. 2017. Establishing transparent IPv6 communication on LoRa based low power wide area networks (LPWANS). In *Wireless Telecommunications Symposium (WTS '17)* (Chicago, IL, USA). IEEE, Piscataway, NJ, USA, 1–6.

[56] Christian Tschudin, Christopher Scherb, et al. 2018. CCN Lite: Lightweight implementation of the Content Centric Networking protocol. http://ccn-lite.net

[57] Valentina Di Vincenzo, Martin Heusse, and Bernard Tourancheau. 2019. Improving Downlink Scalability in LoRaWAN. In *IIEEE International Conference on Communications (ICC'19)*. IEEE, Piscataway, NJ, USA, 7 pages.

[58] Gokcer Yapar, Tuna Tugcu, and Orhan Ermis. 2019. Time-Slotted ALOHA-based LoRaWAN Scheduling with Aggregated Acknowledgement Approach. In *25th Conference of Open Innovations Association (FRUCT'19)*. IEEE, Piscataway, NJ, USA, 383–390.

[59] Tianyuan Yu, Zhiyi Zhang, Xinyu Ma, Philipp Moll, and Lixia Zhang. 2021. *A Pub/Sub API for NDN-Lite with Built-in Security*. Technical Report NDN-0071. NDN.

[60] Lixia Zhang, Alexander Afanasyev, Jeffrey Burke, Van Jacobson, kc claffy, Patrick Crowley, Christos Papadopoulos, Lan Wang, and Beichuan Zhang. 2014. Named Data Networking. *SIGCOMM Comput. Commun. Rev.* 44, 3 (2014), 66–73.

[61] Dimitrios Zorbas, Khaled Abdelfadeel, Panayiotis Kotzanikolaou, and Dirk Pesch. 2020. TS-LoRa: Time-slotted LoRaWAN for the Industrial Internet of Things. *Computer Communications* 153 (2020), 1–10.