Computer Networks 203 (2022) 108656

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

A mobility-compliant publish–subscribe system for an information-centric Internet of Things☆

Cenk Gündoğan [a,*], Peter Kietzmann [a], Thomas C. Schmidt [a], Matthias Wählisch [b]

[a] Hamburg University of Applied Sciences (HAW), 20099 Hamburg, Germany
[b] Freie Universität Berlin, 14195 Berlin, Germany

ARTICLE INFO

Keywords:
ICN
IoT
Producer mobility
NDN
Low-power networking

ABSTRACT

The Information centric networking paradigm has proven particularly useful for the constrained Internet of Things (IoT), in which nodes are challenged by end-to-end communication without network assistance. This work focuses on the interaction between possibly mobile sensors *and actuators* in such IoT regimes which deploy the Named-Data Networking (NDN) architecture. Constrained nodes in interactive scenarios need to be highly responsive but can only manage limited control state. We argue that the request-driven NDN networking paradigm, which prevents pushing of unsolicited data, should be preserved to confine the attack surface, whereas unsolicited *link-local* signaling can accelerate responses without sacrificing security.

In this paper, we contribute HoP-and-Pull (HoPP), a robust publish–subscribe scheme for typical IoT scenarios that targets low-power and lossy wireless networks running hundreds of resource constrained devices at intermittent connectivity. Our approach limits in-memory forwarding state to a minimum and naturally supports producer mobility, a temporary partitioning of networks, data aggregation on intermediary hops, and near real-time reactivity. We thoroughly evaluate the protocol by experiments in a realistic, large testbed with varying numbers of constrained devices, each interconnected via IEEE 802.15.4 wireless LoWPANs. We compare HoPP with common ICN pub–sub and mobility schemes as well as with basic MIPv6 and anchor-based multicast mobility. Implementations are built on CCN-lite with RIOT and support experiments using various single- and multi-hop scenarios.

1. Introduction

The Internet of Things (IoT) is emerging, and billions of new networked devices are forecasted. Currently, a variety of networking technologies are under experimentation for deployment in the low-end IoT. Despite of a maturing IETF protocol suite, dozens of incompatible industry solutions are rolled out to meet device and network constraints, as well as application specific needs [1,2].

Facing this huge world of mainly constrained devices, it seems worth rethinking its networking paradigm. A very loose coupling appears most appropriate between nodes that often run on battery with long sleep cycles and connect via lossy wireless links. Information-Centric Networking (ICN) [3,4] decouples content provisioning from data producers in space which makes it a promising candidate. Additional decoupling in time and synchronization is desirable and attainable by a publish–subscribe layer.

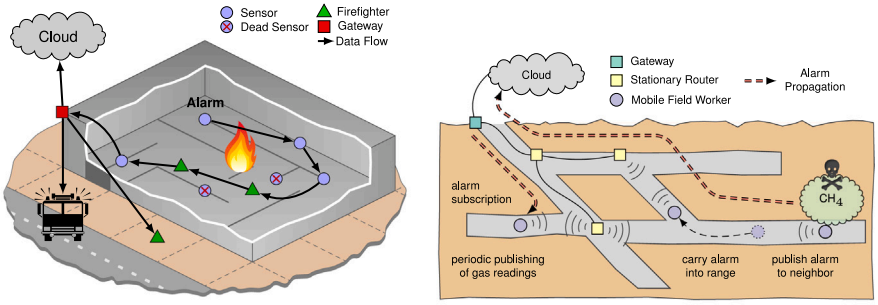
Information-centric publish–subscribe networks have been proposed and an early prominent candidate is PSIRP/PURSUIT [5]. Its central control architecture, however, seems more suitable for an SDN-type deployment in LANs. Publish–subscribe schemes based on NDN like Content-based pub/sub [6] and COPSS [7] violate the loose coupling principle in their use of name-based routing or forwarding. Facing the current state of the art, we explore the problem of information-centric publish–subscribe for IoT networking with a particular focus on mobile and intermittently connected sensors and actuators.

In this paper, we take up the challenge and seek for an information-centric IoT networking solution that qualifies for real-world sensor–actuator deployments with resource-constrained characteristics and low-bandwidth, lossy link properties. We base our work on NDN [8] not only because of its widespread availability and implementations on IoT operating systems, but in particular because of its clean request–response scheme that prevents unwanted traffic at the constrained

☆ This work was supported in part by the German Federal Ministry for Education and Research (BMBF) within the projects *I3 – Information Centric Networking for the Industrial Internet* and *PIVOT – Privacy-Integrated design and Validation in the constrained IoT*.

* Corresponding author.
E-mail addresses: cenk.guendogan@haw-hamburg.de (C. Gündoğan), peter.kietzmann@haw-hamburg.de (P. Kietzmann), t.schmidt@haw-hamburg.de (T.C. Schmidt), m.waehlisch@fu-berlin.de (M. Wählisch).

https://doi.org/10.1016/j.comnet.2021.108656
Received 14 July 2021; Received in revised form 22 October 2021; Accepted 26 November 2021
Available online 13 December 2021



(a) Resilient machine-to-machine communication for the Industrial IoT with heterogeneous constrained IoT devices.

(b) Mobility resilience in harsh environments for the Industrial IoT.

Fig. 1. IoT use cases in industrial settings with device mobility and network partitionings.

end nodes. We present and evaluate HoP-and-Pull (HoPP), a lean, adaptive publish–subscribe layer that strictly adheres to the NDN communication pattern. We extend our previous work [9] by deepening the discussion and considering mobility and support for network disruptions. Our experimental findings on large IoT testbeds indicate that our system complies indeed to the challenging requirements of the IoT use case with promising performance. In particular, reliability and resilience of HoPP largely outperforms previously advised push notifications.

The structure of this paper continues as follows. Section 2 presents safety-critical use cases that challenge current low-power IoT networks by high reliability demands in harsh or mobile environments. We contribute a reflective elaboration of the underlying problem space in Section 3 together with an extensive discussion of related work. In Section 4, we dive into the design details of our publish–subscribe scheme, including the key aspects of network partitioning and publisher mobility. Implementation and evaluations of our system are described in Section 5. Section 6 compares HoPP with common NDN pub–sub and mobility schemes as well as with basic MIPv6 and anchor-based multicast mobility. Finally, we conclude with an outlook in Section 7.

2. IoT use cases

In this section, we focus on two use cases for the deployment of mobile IoT devices in industrial facilities and for safety control in harsh environments.

2.1. Resilient machine-to-machine communication

Industrial settings like oil rigs and warehouse facilities deploy battery-operated devices for collecting sensory data to meet mission-critical requirements and regulatory compliances. Presence detection, smart lighting control, and tracking of exposure to hazards are instances of essential tasks to ensure a safe workplace by mitigating risks to employees and inventory. Replacing or recharging batteries of IoT devices oftentimes incurs high maintenance expenses, especially in confined or hardly accessible spaces. Long battery lifetimes ranging from years to decades are thus desired to minimize deployment costs and are typically achieved with tailored software platforms enabling an optimal power management [10].

Quasi-stationary infrastructures connect these resource-constrained devices to powerful gateways and cloud services using wireless low-power and lossy networks (LLNs) [11]. Notably for regimes with a multitude of IoT endpoints in a single wireless broadcast domain, spuriously disappearing links and saturating network resources are common characteristics. In addition to an energy aware device duty cycling [12], these limitations pose challenges for the host-centric approach to networking, which performs best with perpetual connectivity.

We revisit this basic use case in Fig. 1a, where a timely reporting of incidental threats to industrial settings is decisive for on-site personnel. In this particular scenario the infrastructure is damaged by a fire outbreak, which leaves the network in a partitioned state. Fire fighters and first responders reconnect the network with hand-held devices to receive crucial data on the whereabouts of trapped or unconscious staff members. Seamlessly handling heterogeneous devices and coping with intermittent infrastructure loss are significant qualities of the network in these settings.

2.2. Industrial safety networks

Industrial safety and control systems are increasingly interconnected and operate under harsh conditions. In this use case, we consider industrial environments with a threat of hazardous contaminant (*e.g.,* explosive gas) that need continuous monitoring by stationary, as well as mobile sensors like depicted in Fig. 1b. In case of an emergency, immediate actions are required such as issuing local alarms, activating protective shut-downs (*e.g.,* closing valves, halting pumps), initiating a remote recording for first responders and forensic purposes.

Typical industrial plants are widespread with sparse network coverage, so that mobile workers or machines face intermittent connectivity at scattered gateways. Some sensors and actuators are infrastructure bound, others are independent and battery-powered (*e.g.,* body equipment). The latter resembles the challenges faced in previous DTN-work such as in mines [13].

Like the previous, this use case relies on a fast sensor–actuator network including embedded IoT nodes. In addition, the harsh industrial environment raises the challenges of mobile, intermittently connected end nodes, and network partitioning. Still, enhanced reliability is required in the safety context. We will show in this work, how configurable data replication with dynamically generated content proxies can meet these challenges and how they combine in a lightweight system suitable for real-world deployment [14].

3. The problem of information centric IoT networking and related work

3.1. Device mobility and network disruptions

Mobile nodes are part of many IoT deployments. While mobility is natively supported at the receiver side of NDN, publisher mobility is considered difficult to solve in a generic way [15]. Translated to IoT use cases, this means mobile sensors are hard to integrate—a particular problem for surveillance and safety sensing applications [14]. Related scenarios may also experience temporary network partitioning, which can be treated with correspondence to network mobility.

KITE [18] takes a soft-state approach where mobile producer nodes proactively build temporary paths (*traces* [16]) to a rendezvous point

Table 1
Overview of the related work grouped according to the main contributions.

Category	Characteristics	References
Routing and mobility	Use cases & surveys	[15–17]
	Producer mobility	[18–20]
	Locator/identifier split	[21–24]
	Delay-tolerance	[25,26]
	Opportunistic routing & signaling	[27–30]
	Stateful & adaptive forwarding	[31–34]
Publish–Subscribe	Rendezvous-based delivery	[7,35–38]
	Dataset synchronization	[39]
	Semantic naming & routing	[40–42]
Information-centric Internet of Things	Architectures & deployment reports	[2,14,43–49]
	Security threats & analyses	[50–53]
	Link-layer extensions	[54–56]
	Unsolicited data delivery	[6,57–59]

as soon as they move or anticipate data retrievals from consumers. Consumer traffic generally traverses the rendezvous servers, but the hop-by-hop forwarding of NDN allows to reduce path stretch for consumers that share parts of the path with a mobile producer.

An alternative suggestion that handles an anchor-less producer mobility is provided by *MAP-Me* [20]. In this extension, mobile producers send special Interests to name prefixes they own in order to update obsolete forwarding states. With the premise that an underlying routing protocol operates much slower, such Interests traverse to the old locations of recently moved producers. Any recipient of a special Interest then updates its forwarding information base by recording the incoming face alongside the name prefix inside the Interest.

A recent publish–subscribe system with consumer and producer mobility support [19] uses persistent PIT entries to serve multiple Data packets along a request path. A practical cleanup routine reveals the absence of expected data (*e.g.,* due to mobility) and tears down unnecessary soft-state in the network. Data packets carry infrastructure-specific information to detect routing inconsistencies and to trigger routing repairs.

Other approaches separate human-readable content object names and network address locators in order to handle producer mobility—a concept that is part of the MobilityFirst [21] design considerations and is also adopted for NDN based architectures [22–24]. Solutions that use a controlled flooding of Interests and broadcast mechanisms of the link-layer [27–29] show less infrastructure requirements, but generally produce more signaling overhead.

Systems that act on the information-centric maxim already exhibit a decoupling of data and location and therefore potentially qualify for deployments with long network disruptions. Delay tolerance for NDN can be enhanced by integrating the Bundle protocol [25], or with specific content caching mechanisms [26].

3.2. Naming and routing

Naming content on an information-centric network layer promises a simplified access to information. Routing on names directly designs a lean network without further address mapping. It obsoletes infrastructure like the DNS and eliminates the attack surface inherent to the mapping. Both aspects are of great advantage in a constrained IoT network. However, name-based routing encounters the problems of (*i*) exploding routing tables, as the number of names largely exceeds common routing resources, and (*ii*) limited aggregation potentials, as names are specific to appliances and applications, but independent of content locations. More severely and in contrast to IP, a local router cannot decide on aggregating names since the symbol space of names is not enumerable in practice [33]. Limiting the complexity of name-based routing and FIB table state is one of the major challenges in IoT networks [17].

Routing normally proceeds according to location information from the FIB. Names in FIBs only aggregate well if naming follows the topological hierarchy of the network. This rarely holds, since naming is application-specific, and cannot be detected without distributed knowledge. To overcome FIB explosion, several authors refer to the NDN capabilities of stateful forwarding, using the option of distributing requests to several interfaces simultaneously [31,32]. Such Interest multicasting will lead to duplicate content deliveries if the network is densely connected. In 'Pro Diluvian' [30], the effects of such scoped flooding are analyzed, and authors find a utility limited over very few ($\approx$ 2–3) hops. Such opportunistic forwarding can also lead to loops, as was pointed out by Garcia-Luna-Aceves [34]. In any case, the excessive traffic, as well as redundant PIT states make this approach infeasible for the IoT.

COPSS [7], an earlier publish–subscribe approach inspired by PIM [35] multicast routing, selects a rendezvous point to interconnect publishers and subscribers. Such dedicated routing point naturally allows for name aggregation. Like PIM-SM (Phase 2), COPSS further establishes a dedicated forwarding infrastructure (subscription table) that establishes persistent forwarding paths from the publisher via the rendezvous point to the receivers.

A publish–subscribe framework with a focus on building management systems (ndnBMS-PS [39]) uses the functionality of repositories to publish data following signed command Interests from producer nodes. Repositories then replicate in the network and content synchronizes to subscriber nodes using a synchronization protocol for NDN. In contrast to ndnBMS-PS, which requires an external topology management, the publish–subscribe Internet (PSI) [36] architecture provides the two network primitives publish and subscribe, as well as topology managers for path computations between host nodes. Similar to COPSS, PSI uses rendezvous points to match announcements and subscriptions. TPS-CCN [42] is a topic-based publish–subscribe CCN system that integrates a MANET link state routing protocol to identify available topics in the network. The naming scheme includes the topic prefix and appends a version number to track the evolution of content for a specific topic. Subscribers then request content objects using standard Interest messages for names with progressively increasing version numbers. In case of network disruptions, a delay-tolerant mode allows for broadcasting Interests to explore the close vicinity for desired content.

MFT-PubSub [37] builds a spanning tree on an IP network overlay using a leader election algorithm. Subscriptions propagate along the tree topology to all brokers and are recorded in the corresponding routing tables. Announcements are then forwarded to subscribers according to the existing forwarding state. On network partitioning, local leaders are elected to maintain the routing infrastructure in each partition. Other approaches either organize topics under common prefix trees [40,41] to rely on the prefix matching capabilities performed by the NDN forwarding fabric, or directly utilize Interest messages to push data towards subscribers [38].

PANINI [33] re-uses the idea of an aggregation point called Name Collector, but does not establish a (persistent) forwarding plane like COPSS. Instead, PANINI uses selective broadcasts to discover unpopular routes towards the network edge. For the IoT, we want to minimize control traffic and avoid flooding. We restrict our solution to a lean default routing, instead.

3.3. ICN in the IoT

It became apparent [2,43,44] that ICN/NDN exhibit great potentials for the IoT. Not only allows the access of named content instead of distant nodes a much leaner and more robust implementation of a network layer, but in particular prevents the request–response pattern of NDN any overloading with data at the receiver.

3.3.1. Security, resilience, and robustness

For a few years, it was the believe that NDN can be DoS resistant by design, until Interest- and state-based attacks were discovered [50]. Subsequent work [51,52] elaborated the threats of Interest flooding and overloading FIB and PIT tables by user-generated names and content requests. This has proven difficult to mitigate [53] and is a particular threat to memory-constrained nodes. In the subsequent Section 4, we will show how a FIB with simple default routes can serve the IoT, and how PITs remain minimal by hop-wise content replication between nodes.

ICN deployment in the IoT has been studied with increasing intensity, touching protocol design aspects [45,46,55], architecture work [47–49], and practical use cases [14,57,60,61]. Emerging link-layer extensions for the wireless like TSCH turned out to be beneficial for the interaction of NDN communication patterns and channel management [54]. Several implementations have become available. CCN-Lite [62] runs on RIOT [63] and on Contiki [64], NDN has been ported to RIOT [65]. Thus, grounds seem to be prepared for opening the floor to real-world IoT applications with NDN.

Many deployments in the IoT, though, follow the communication patterns *on demand*, *scheduled*, and *unscheduled*. Actuators in particular rely on unscheduled control messages. Since NDN is built on the request–response scheme of data-follows-Interest, unscheduled push messages are not natively supported. For the IoT, this has been identified as a major research challenge [17].

3.3.2. Push communication

Several extensions have been proposed to enable an unsolicited push of data, among them *Interest-follows-Interest* [57], *Interest notification* [58], and a dedicated *push packet* [59]. All these push messages are sent immediately to a prospective consumer node, which not only conflicts with the ICN paradigm of naming content instead of hosts, but has no forwarding supported on the network layer. No push packet will reach its destination unless potential receivers are announced to the routing using a node-centric name. Unidirectional data push to named nodes, however, lacks flow as well as congestion control, and opens an attack surface to DoS. In the IoT with its constrained nodes, this must be rated a particularly severe disadvantage.

Carzaniga et al. [6] with a proposal of *long-lived Interest* seem to be the first in addressing the push challenge in a natural NDN fashion. Subscribers issue a persistent Interest that is not consumed at content arrival, and thereby establish a (static) data path from the producer. Unfortunately, long-lived Interests open an unrestricted data path to the recipient and thereby inherit the threats of overload as other push primitives. In addition, persistent forwarding states in PITs lead to self-reinforcing broadcast storms whenever L2 broadcasts are used [56]. Finally, frequent topology changes as characteristic for the IoT will routinely break paths. In the following, we will show how regular Interests with appropriate lifetime can serve this purpose equally well, without suffering from its drawbacks.

3.4. Requirements for an ICN-based IoT publish–subscribe system

Typical IoT scenarios impose critical requirements on a publish–subscribe system. The state of the art as summarized in Table 1 mainly focuses on general purpose deployments with sufficiently provisioned network and device resources. We derive three challenges from the related work, which need to be addressed for creating a robust and energy frugal content replication mechanism. First, IoT deployments may consist of hundreds of resource constrained devices with intermittent connectivity. A publish–subscribe approach for these setups needs to minimize forwarding states as they scale with the number of network participants. While a powerful gateway device can potentially hold enough in-memory forwarding information to represent the whole network, the constrained devices have rather limited memory space for forwarding states. Second, consumer and producer mobility as well as temporary network partitionings are prevalent and must be handled by any publish–subscribe solution for the IoT. Regular packet loss is expected and corrective actions without an excessive overhead is required, while not inhibiting the protocol reactivity. Third, the constrained processing and memory resources of common IoT hardware necessitate a low implementation complexity. All applications on an IoT software platform compete for available resources and wasteful uses limit the device operability.

In Section 4, we design a robust publish–subscribe system that meets these requirements to enable a resilient content replication. A real protocol implementation for an IoT operating system demonstrates the feasibility of the described approach.

4. HoP and pull: A publish–subscribe approach to lightweight routing on names

4.1. Overview

We now describe HoP-and-Pull (HoPP), our pub–sub system for lightweight IoT deployment. For a confined IoT environment, we make the common assumption that nodes form a stub network that may be connected to the outside by one or several gateways. Some global prefix is given to a gateway, but (wireless) IoT nodes can reach a gateway without global prefix changes in one or several hops unless they are temporarily disconnected [66]. Internally, nodes may be grouped according to one or several sub-network prefixes (*e.g.,* `/valves`).

We select one or several distinguished nodes to serve as Content Proxies (CPs) on infrastructure setup time. CPs are typically more stable and more powerful than the constrained edge nodes. The CP function may reside on gateways or other infrastructural entities of a deployed system. These CPs take the role of data caches and persistent access points. They will be reachable throughout the network by default routes, unless temporary partitioning occurs. Note that one CP can serve several local prefixes, but a local prefix may also belong to several CPs. The latter scenario will lead to replicated caching with higher and faster data availability.

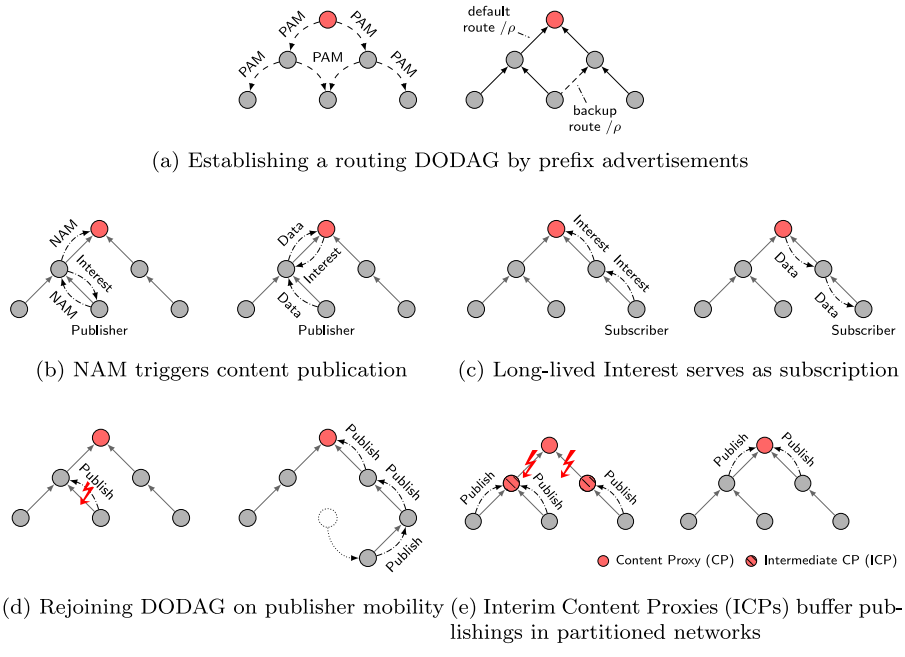
Our publish–subscribe protocol for the IoT is then composed of three core primitives:

1. Establishing and maintaining the routing system
2. Publishing content to the Content Proxies
3. Subscribing content from the Content Proxies

Our following protocol definition strictly complies with the design principles: (a) minimal FIBs that only contain default routes, (b) no push primitive or polling, (c) no broadcast or flooding on the data plane. The HoPP protocol transparently manages consumer and *producer mobility* as could be demonstrated in our prototype [67].

4.2. Prefix-specific default routing

Content Proxies advertise the prefix(es) they own on the control plane to all direct neighbors in a Prefix Advertisement Message (PAM). PAM messages are link-local, and do not interfere with regular NDN network operations. This orthogonality leaves the primary data structures Pending Interest Table (PIT) and Content Store (CS) unaffected. Hence, route signaling performs a topology discovery on a strictly scoped and shielded control plane. Observing nodes will adopt a CP as their parent and re-distribute the PAM message to their neighbors with an increased distance value. Much like in the core RPL [68], parents broadcast PAMs to the one-hop vicinity, which allows for an increased scalability independent of the neighborhood size. Trickle [69] regulates the rate of message transmissions to substantially reduce the broadcast chattiness in stable network topologies. All nodes will become members of a Destination-Oriented Directed Acyclic Graph (DODAG) while routing converges. Any topological change, *e.g.,* due to mobility



(a) Establishing a routing DODAG by prefix advertisements

(b) NAM triggers content publication

(c) Long-lived Interest serves as subscription

(d) Rejoining DODAG on publisher mobility

(e) Interim Content Proxies (ICPs) buffer publishings in partitioned networks

Fig. 2. Overview on the HoPP protocol operations: topology management, publish–subscribe, mobility and delay-tolerance support.

or parent timeouts, resets the Trickle algorithm to quickly trigger PAM announcements and, thus, converge towards a consistent DODAG with refreshed forwarding states, before reducing the chattiness again.

Nodes select the best seen uplink in their FIB as default route to the announced prefix, but may add additional uplinks with lower priority for backup. The selection process uses the hop-count metric, but also allows for the integration of more sophisticated alternatives, *e.g.,* MRHOF [70].

A deployment always includes at least one CP serving a specific name prefix. If multiple CPs announce the same prefix, then nodes configure multiple default routes for this particular prefix. Unlike in host-centric deployments, NDN inherently supports a multi-destination forwarding and thus enables a seamless data replication onto several CPs.

Fig. 2a visualizes the PAM prefix distribution and the corresponding FIB entry for the sample prefix $/\rho$. All nodes establish prefix-specific default routes on their shortest paths upstream. In addition, nodes learn backup paths of equal hop distance, which may be of lower radio quality.

4.3. Publishing content

An IoT node (sensor) that has new data to publish will first select a name. It may choose either from a predefined scheme accessible by local controllers, some common standard set, or decide individually. Since generated names are expected to be unique across the whole system, they include device-specific identifiers to partition the naming scheme. It is typical for time series sensor data to append an increasing counter or the current timestamp to a name prefix. If the uniqueness of names cannot be guaranteed within the stub network, then a duplicate name detection is necessary. The specific method is out of scope of this document, but the multihop duplicate address detection (DAD) of the neighbor discovery protocol (NDP) [71] provides a viable base.

It will advertise this content name to its upstream neighbor via a (unicast) Name Advertisement Message (NAM). It will also associate the content with one or several topic names and adds these to the content metadata. Depending on the publishing rate of content, a node can announce multiple names in a single NAM message. This aggregation of names is however limited by the maximum transmission unit (MTU) of the underlying link-layer.

Under regular network conditions, the upstream neighbor is expected to retrieve the advertised content via the incoming interface of the NAM. It proceeds according to the standard NDN scheme: An Interest requests the name, the data is returned in response. Concurrently, the upstream issues a corresponding NAM to its parent, which in turn pulls the content one hop closer to the CP. This hop-wise content replication proceeds until the data arrives at the CP.

It is worth noting that the NAM content alerting is situated on the control plane using *link-local unicast* signaling. Neither a data path is established in the PIT, nor are FIBs modified. NAM content signaling also complies with the strictly scoped and shielded control plane of HoPP.

The publishing mechanism is depicted in Fig. 2b. A Publisher issues a NAM to its parent, which requests the content and republishes the NAM towards the CP in parallel. The content request is performed on the NDN data plane via a regular interest-data handshake. If a single NAM includes multiple name announcements, then each of the name is requested separately. After arrival of the data, nodes satisfy outstanding Interests up to the CP.

At irregular network conditions, a node may not receive an Interest that matches its previous name advertisements. This may be due to broken links, failing or deep-sleeping nodes, or enduring overload. After a deployment-specific timeout, the content owner will adapt and try to publish the content on an alternate path by sending a NAM up on a backup link. In case of a complete failure, the content node can follow two strategies: Either it waits and re-advertises according to an exponential back-off, or it solicits a refresh of router advertisements for learning new, operational routes. In the latter case, nodes send multicast SOL (solicitation) messages to trigger PAM messages from immediate neighbors.

4.4. Subscribing to content

A subscriber in HoPP behaves almost like any content requester in NDN. It issues a regular Interest request up the default route to the CP and awaits the response. There are two deviations from plain NDN, though. First, the subscriber cannot extract content names from its FIB, since FIBs only contain prefixes. These prefixes, however, can serve as topics in the context of confined application deployment. Second, to meet real-time alerting requirements of publishers, a subscriber can

issue timely Interests with extended lifetimes to immediately receive published content once it arrives at the content proxy.

Names are expected to follow an application-specific logic. In a publish–subscribe system, individual names of content items are grouped according to topics, which itself appear as prefixes in the naming hierarchy. The corresponding CP will answer the request for a topic with an empty data chunk that carries available content name(s) as metadata, *e.g.,* in a manifest [72].

Fig. 2c displays the operations of a subscriber. An Interest for named content is sent up to the proper prefix owner (CP) and remains for a predefined lifetime, if the Content Proxy cannot supply the data. These requests terminate at the CP anchor and do not propagate downwards to the actual publishers. In case content is arriving from a publisher to the CP, data is transferred automatically down the reverse Interest path—as a regular NDN operation. We anticipate that in common sensor–actuator networks of the IoT, the application semantic will define meaningful Interest lifetimes. Otherwise, in regimes of largely fluctuating temporal behaviors or long-lasting subscriptions (*e.g.,* alerts), the subscriber may refresh and maintain the request at its discretion.

Note that in contrast to *long-lived Interests* or the COPSS *subscription tables*, such Interests of extended lifetime are consumed by arriving content and do not open a persistent, uncontrolled data path. Subscribers continue to apply flow control and may discontinue subscriptions to unwanted content.

4.5. Publisher mobility and network partitioning

A publishing node that moves from one point of attachment to another within the IoT domain, will experience stable routing conditions in the sense that default routes to active prefixes should exist everywhere in a connected network. Correspondingly, the mobile node (MN) can re-configure its upstream route either by waiting for the next prefix advertisement (PAM), or may actively solicit an additional PAM to discover new, reachable parents. Note that these link-local route configurations are of low complexity and closely resemble the autoconfiguration of IPv6 default gateways. In contrast to Mobile IPv6 [73], though, the MN in our publish–subscribe system can continue publication immediately after a link-local route is re-established by a newly arriving PAM as outlined in Section 4.2 for building and maintaining the topology. This makes the handover process lightweight and very fast.

Fig. 2d illustrates provider mobility. A publisher removes from the network while trying to publish a content item and enters the radio range of another node in the DODAG. It may now actively learn about network re-attachment (*e.g.,* from link triggers), or learn from a newly arriving PAM. After the local upstream is configured, the mobile publisher can successfully complete its publishing handshake.

Temporary network partitioning proceeds very similar to mobility. An intermediate node that looses upstream connectivity will explore alternate paths (*cf,* Section 4.3), but has to await a re-attachment in case of a complete failure. Such node will continue to receive publishing demands (NAMs) from the downstream, which it will satisfy in accordance with its resources. On overload, it will terminate to retrieve content from its children. Proceeding this way will establish a classic backpressure mechanism of flow control.

Operations under network partitioning are shown in Fig. 2e. Following an outage of the CP, immediate children experience a disconnect. Forwarders act automatically as Interim Content Proxies (ICPs) once they lose upstream connectivity. ICPs are temporary content proxies, and they store all published content as long as they have enough buffer resources. They continue to handle publications (as well as subscriptions) until connectivity to the CP is re-established, in which case a forwarder re-publishes all delayed data to the newly chosen upstream parent.

5. Implementation and evaluation

5.1. Implementation for CCN-lite on RIOT

We implemented the HoPP extensions on the CCN-lite version ported to RIOT and deploy NDN. It is noteworthy that this software stack supports both, the NDN core protocol as well as CCNx. On RIOT, CCN-lite implements the `netdev` interface and runs as a dedicated single-threaded network stack.

The architecture of HoPP is depicted in Fig. 3. It mainly adds a new control protocol block that handles exchange and processing of the two new packet types (PAM, NAM) on the control plane. This extends the `forwarder` module of CCN-lite. The `forwarder` allows extensions for the packet parsing by the use of user-defined callback functions on a suite basis. Considering this loose coupling, the actual topology maintenance was implemented separately from the CCN-lite core. The `topology manager` handles PAM scheduling and parent selection to form and maintain the routing topology (DODAG). Resulting forwarding states are reflected in the FIB with the help of the CCN-lite API. The Name Advertisement Daemon (`NAD`) module handles parsing and scheduling of NAM messages. A NAM Cache (`NC`) is used to intermittently track the hop-wise propagation and to reschedule NAM transmissions in case of network disruptions. For each entry in the `NC`, the `NAD` triggers the `replicator` to invoke a hop-wise content replication on the data plane via pull-driven Interest-Data. To ensure hop-wise replication of published content, a caching strategy was added to CCN-lite that hinders replicated content to be cached out during publishing. After a successful Interest-Data exchange, the `replicator` notifies the `NAD` module and the appropriate `NC` entry is freed for removal. We note that the newly added data structure (`NC`) is very lightweight, since it only tracks the names of unpublished content objects. The content itself resides in the default CCN-lite Content Store (CS). The NAM Cache is orthogonal to the other data structures and is implemented outside of CCN-lite within the NAD. HoPP maintains the CCN-lite FIB data structure, but does not interact with the Pending Interest Table (PIT) and CS. The latter structures are regularly updated by the normal NDN operation through Interest and Data messages.

5.2. Experiment setup description

All experiments are conducted in the FIT IoT-LAB testbed [74] to reflect common IoT properties. The testbed consists of several hundreds of class 2 [75] devices equipped with an ARM Cortex-M3 MCU, 64 kB of RAM and 512 kB of ROM, and an IEEE 802.15.4 radio (Atmel AT86RF231). The radio card provides basic MAC layer functions implemented in hardware, such as ACK handling, retransmissions, and CSMA/CA. The software platform is based on RIOT [63] and an extended CCN-lite network stack.

The performance of the HoPP publish–subscribe IoT system is evaluated on the three different topologies:

Paris is a densely connected topology of 69 nodes all within radio reach.

Grenoble (ring) is formed of a closed rectangle with two double-stacked edges. 178 nodes form a heterogeneously meshed network with a maximal hop distance of four.

Grenoble consists of about 350 nodes, where half of them is situated on the rectangle, the other half forms linear extensions leading outwards. This network supports complex topologies with a node distance up to 9 hops.

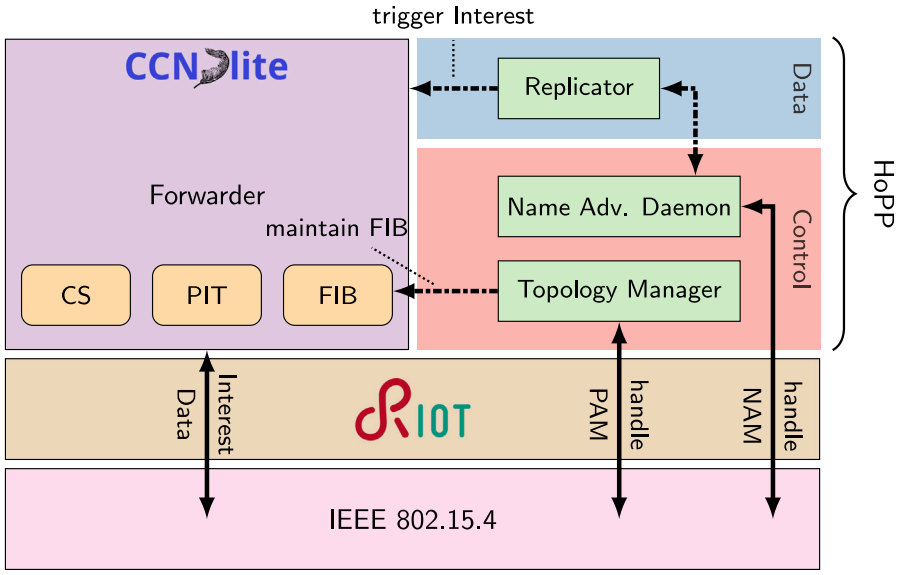


Fig. 3. IoT publish–subscribe architecture.

While the physical location of each stationary node is fixed by the actual testbed infrastructure, we carefully select devices that are sufficiently apart from each other to promote the logical formation of meshed multi-hop topologies. Since all nodes are within broadcast range on the Paris site, they always connect to the HoPP content proxy to form a star topology. On the other hand, the resulting meshed topologies for the Grenoble site have many branches with long path stretches. With these topologies, we model the aspects of typical IoT use cases: star topologies are usually deployed in scenarios where devices stay in proximity of a single base station with uplink connectivity. Remote deployments with mobile handhelds in challenging environments, *e.g.,* in confined spaces as illustrated in Section 2, use meshed multi-hop topologies.

We illuminate multiple protocol aspects throughout the following sections using the three selected topologies. In Section 5.3, we measure the success rates for publishing content as it is an important metric to gauge the efficacy and scalability of this protocol in lossy environments. To assess the reactivity of the pull-driven HoPP approach in these low-power regimes, we compare it against a push-based protocol. Since HoPP also builds and maintains a logical routing topology, we further measure the routing convergence time for the three selected site formations. Section 5.4 provides further insights on the resiliency of protocol operations in partitioned and mobile networks.

5.3. Evaluation of the HoPP baseline performance

The first evaluation inspects the reliability of HoPP compared to the plain Interest notification [58] approach, which allows to encode unsolicited data in request messages. We investigate the content reception rate on a given consumer in the Grenoble ring multi-hop topology using a converge cast traffic pattern, where each device generates sensor readings every 30 ± 15 s.

Fig. 4a compares the reliability of HoPP with the common Interest Notification approach in relation to the hop distance of the consumer. For HoPP, we observe a steady high content delivery rate above 96% for all hop distances in the topology. NDN Interest Notification admits significantly lower reliability and shows a decline in transmission with increasing hop distance. While a hop count of 1 yields 70% packet arrivals, success ratio decreases to 41% for hop distances of 5 and larger. Next, we investigate performance metrics that relate to the temporal behavior of the protocol. Since deficits of the core protocol, but also different failures of networked elements (radio/link layer, CCN-layer, pub–sub, and node layer) translate into delays due to retransmissions and re-arrangements, times to completion are a key performance indicators. In detail, we study (*i*) routing convergence, (*ii*) times to publish content items, (*iii*) times to publish under network partitioning, and (*iv*) times to issue alerts (from publisher to the subscribers).

Routing convergence times in the three testbeds are displayed in Fig. 4b. Clearly visible is the dependence on hop counts, each counting for an average delay of $\approx$ 100 ms—the PAM timer. While Paris is a single-hop network and exhibits a single step in distribution, multiple steps represent hop count multiplicities in the multi-hop cases. No exceptional delays become visible. This is due to the moderate timing of the routing protocol which causes a low network utilization.

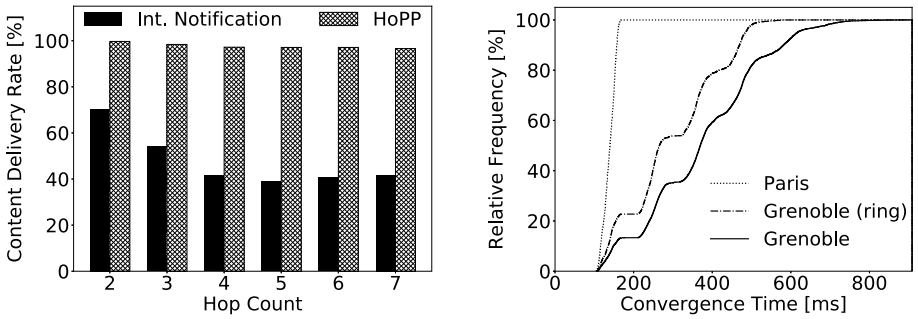
For the evaluation of the times needed to publish a content item, we iterate the following. For each topology, a Content Proxy is positioned in the center of the network, while randomly chosen nodes publish a single, individually named chunk to the network. Publication is initiated every 30 ± 15 s and depending on the nodes position in the tree, one to several data packets traverse the same sub-paths within very short time frames.

Results for the single-hop network (Paris) are displayed in Fig. 5. Observing round-trip ping values of $\approx$ 10 ms, the NAM timer ($\texttt{nam}_\texttt{t}$) of 125 ± 25 ms, and the CCN-lite processing, a mean time to publish of about 135 ms would be expected. Small fluctuations at $\approx 2 \times \texttt{nam}_\texttt{t}$ indicate additional delays that result from network disturbances and node congestion leading to paths of hop count two.

Similar results become visible from the Grenoble experiments in Fig. 5. Clearly pronounced are the first four routing hops, higher hop counts blur according to increasing fluctuations for the Grenoble topology. Roughly 80% of all publish events complete below one second in the Grenoble (ring) topology, while a similar amount of events require up to two seconds for the Grenoble setup. These results clearly show the fragility of the lossy wireless regime, but also confirm a majority of these challenging transmissions did complete on the expected time scale.

Now, the end-to-end delay from the publisher to the subscriber is examined. This corresponds to the use case of issuing alerts between nodes from the local IoT network. In addition to the publish events of the previous measurements in Fig. 5, this scenario also includes periodic subscription requests that are issued randomly scattered within the topology at intervals of $\approx$ 30 s.

The experimental output for the three topologies are displayed in Fig. 6. As we might expect, blurring fluctuations have enhanced with



(a) Success rate of content delivery to one consumer as a function of hop count in the Grenoble multi-hop testbed

(b) Routing convergence time for the testbed topologies

Fig. 4. Evaluation of success rates and convergence time.

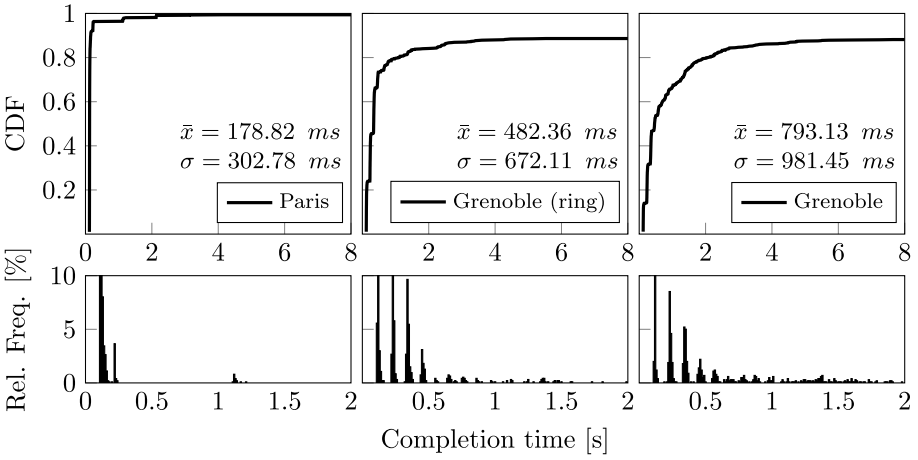


Fig. 5. Time to content publishing for multiple publisher nodes towards the Content Proxy.

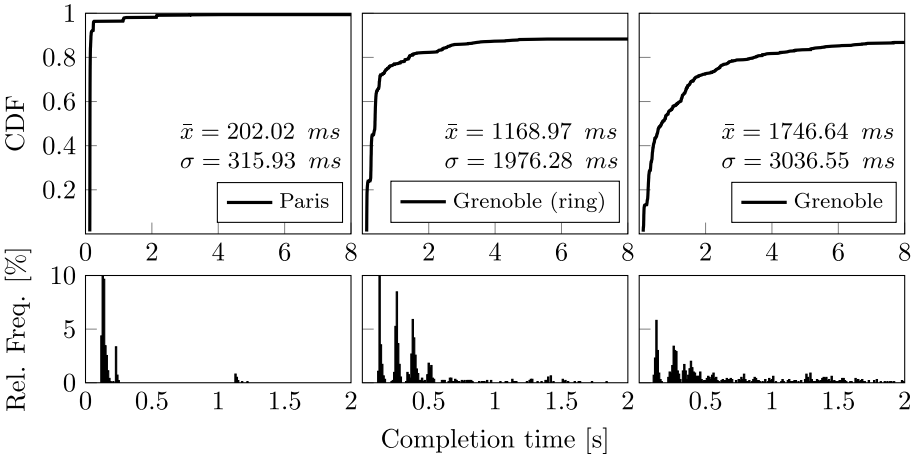


Fig. 6. Time to issue alerts from publisher nodes to subscribers via the Content Proxy.

only a few pronounced signatures of hops and the means increased slightly by the extended paths towards the subscribers. Notably, the single-hop testbed from Paris performed best under the extended communication load, whereas the full Grenoble testbed clearly runs at its limit. The latter can be easily explained by the many hop transitions required at Grenoble, each of which requires an additional packet exchange which potentially impacts on neighbors within radio range.

Low power lossy networks that connect heavily constrained IoT nodes are known to be infeasible for such heavy load. We consider it a success that a notable fraction of the content arrived at its receivers on within about 500 ms—a timescale which is considered normal in multi-hop WPANs. To a certain degree, we account this for the robustness of our hopwise content publishing and replication protocol. Further evaluations [2] confirm these observations.

Finally, we inspect the network overhead of publish operations for the three selected deployments. HoPP publishers replicate content hopwise to parent nodes until they arrive at the content proxy. During this process, the actual data packets follow a sequence of NAM and Interest messages, *i.e.,* in the optimal case without any packet loss, the network overhead on a single link per successful content transfer consists of

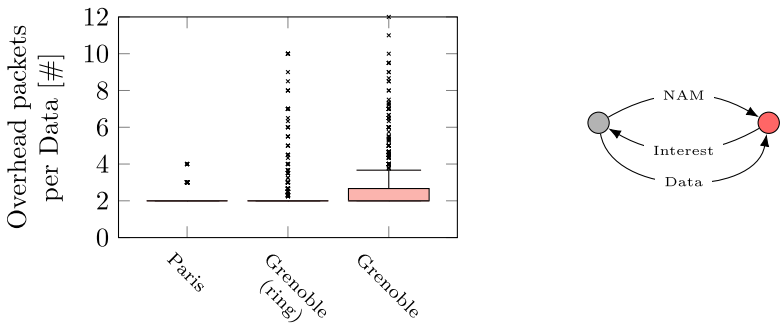


Fig. 7. Overhead packets per publish event normalized by the hop count of a publisher for the three protocol deployments. The optimal overhead packet count is two (NAM and Interest).

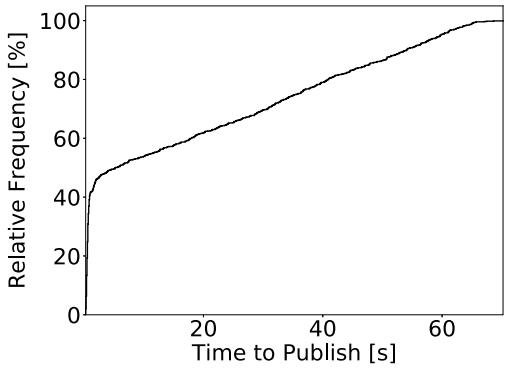


Fig. 8. Time to content publishing at network partitioning.

two messages. On packet loss, *e.g.,* due to saturated link resources or wireless interferences, the network overhead increases as the NAMs and Interests are retransmitted.

We count the number of distinct packets for all publish operations across all nodes in a specific topology, normalize them by the hop count for each node, and display the statistical key properties in Fig. 7. While we observe a median of two for all three topologies, the statistical dispersions show noticeable differences among the varying deployment options. As already observed in the former evaluations, the Paris topology experiences the least network stress and hence shows an overhead distribution centered around the median of two without any virtual variability. A few outliers indicate an increased overhead for two publish operations. The Grenoble (ring) topology increases in node density and path stretch, which marginally impacts the overhead: the variability around the median is still minimal, but more outliers indicate additional retransmission events. For the Grenoble deployment, we observe an enlarged spread where the middle 50% of measured data reaches an overhead of two to three packets. The outliers considerably increase, which signifies more network stress and is hence in line with the previous protocol evaluations.

5.4. Performance evaluation of mobility & network partitioning

We analyze a scenario of network partitioning on the Grenoble ring topology. To quantify the effects of a major network disruption, we disabled all nodes that are two hops away from the Content Proxy every 60 s for an off-time interval of 60 s. This isolated the Content Proxy periodically. Content publishing proceeded randomly with a frequency of one per 30 ± 15 s.

Results in Fig. 8 highlight a smooth content transition to the CP with a timing almost linearly stretched over the 60 s off-period. No unexpected content delays become visible, which indicates the protocol robustness on this macroscopic time scale.

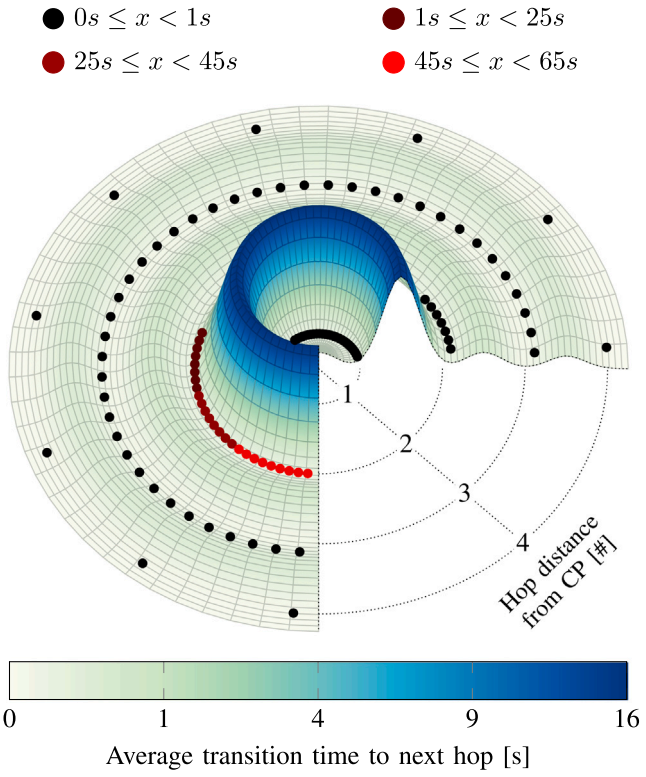


Fig. 9. Visualization of publish events and publish time between hops in a partitioned network. Rings denote the hop distance of publisher nodes to the Content Proxy, which is situated in the center. Dots represent publish events originating from a node on a ring towards a parent node (next inner circle). Wave amplitudes and color codes of dots indicate the publish time to the next hop (black is quickly published, red has long buffer periods). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The illustration in Fig. 9 provides a more detailed overview on the number of publish events per hop and the time needed to replicate content objects to the immediate parent node. For an enhanced visualization, a single dot in Fig. 9 represents ten publish events during the duration of the experiment. Each dot is situated equidistantly on a ring, which symbolizes the distance of the publisher to the proxy node, *e.g.,* four hops for the outermost ring. Inner circles show more events due to the accumulative nature of publishing towards a single root node in the tree topology.

Waves between rings indicate the average duration of a content to be replicated to the next hop. The publish operation consists of the three messages NAM, Interest, Data, and needs 200–500 ms to finish for the majority of the events between rings 4→ 3, 3 →2, and 1→root. Similar numbers were also recorded in our previous measurements

Table 2
Theoretical space complexity analysis, where $\mathcal{F}$ denotes the number of forwarding entries (FIB) and $\mathcal{P}$ denotes the number of pending requests (PIT).

Protocol	Forwarding state requirements					
	Consumer		Producer		Forwarder	
	FIB	PIT	FIB	PIT	FIB	PIT
NDN	$\mathcal{O}(\mathcal{F})$	–	–	$\mathcal{O}(\mathcal{P})$	$\mathcal{O}(\mathcal{F})$	$\mathcal{O}(\mathcal{P})$
HoPP	$\mathcal{O}(1)$	–	$\mathcal{O}(1)$	–	$\mathcal{O}(1)$	$\mathcal{O}(\mathcal{P})$
PubSub-Mob [19]	$\mathcal{O}(\mathcal{F})$	–	–	$\mathcal{O}(\mathcal{P})$	$\mathcal{O}(\mathcal{F})$	$\mathcal{O}(\mathcal{P})$
NDN-Lite Pub–Sub [40]	$\mathcal{O}(1)$	–	–	$\mathcal{O}(1)$	–	–

(Fig. 5). While waves between these rings show the same amplitude, an exception occurs between hop two and hop one: due to the configured partitioning, we observe much higher replication times, as already seen in Fig. 8. ≈50% of all events between rings 2→1 show timings that increase from a few seconds up to 65 s. This gradual increase is depicted in the color coding of the dots. Darker events suggest a sub-second replication time, whereas red events indicate times up to a minute, in which case they are buffered for longer periods until the network reconnects.

6. Protocol comparison

In this last part, we compare the memory requirements and incurred signaling overhead of HoPP with alternative publish–subscribe and mobility approaches.

6.1. Qualitative memory assessment

Varying publish–subscribe solutions show different memory requirements to store and maintain forwarding states on producer, consumer, and forwarder nodes. Since main memory is scarce on typical class 2 devices, exhaustive schemes greatly impact the deployment scalability. In this comparison, we theoretically assess the state space for different publish–subscribe protocols (see Table 2).

First, we analyze necessary states to forward Interests (FIB) and Data (PIT) using the pure NDN protocol as baseline. The most decisive part affecting RAM usage is the name component: a hierarchical, descriptive naming scheme may include hash-based device identifiers, key fingerprints to handle access control, and timestamps to denote content recency. Requiring around 100 bytes per forwarding entry including face information like next-hop hardware addresses is therefore not unusual. For global producer reachability, each consumer maintains forwarding rules in the FIB commonly installed by an orthogonal routing protocol. While a highly hierarchical naming scheme may allow forwarding states to aggregate, space demands increase linearly with the number of published names ($\mathcal{O}(\mathcal{F})$) in the worst case. In contrast to consumers, producers do not require additional forwarding state due to the reverse path forwarding logic. Forwarders maintain reachability state for each name ($\mathcal{O}(\mathcal{F})$) and further maintain soft-state for each pending Interest ($\mathcal{O}(\mathcal{P})$) to preserve reverse paths. The linear increase in $\mathcal{F}$ and $\mathcal{P}$ is again most distinct in scenarios where state aggregation is not possible.

HoPP operates in lossy networks consisting of low-end IoT devices and decouples producers from consumers by using CPs as anchor nodes. As a consequence, HoPP requires only a single forwarding entry that points towards a CP on all devices with constant memory use ($\mathcal{O}(1)$). Additionally, forwarders maintain short-lived PIT state for subscriber Interests, which linearly increases with the number of pending requests ($\mathcal{O}(\mathcal{P})$). The number of available content publishings and subscribers does not affect the state requirements on producers and consumers, which is a necessary requirement for large-scale deployments.

Forwarding state in PubSub-Mob [19] is maintained and distributed via the external routing protocol NLSR [76]. Consumers and forwarders exhibit similar memory needs as a pure NDN deployment. Since this alternative publish–subscribe mechanism makes use of persistent PIT entries on producers, these nodes additionally store soft-state for the duration of each subscription event, *i.e.,* pending Interest.

NDN-Lite Pub/Sub [40] assumes all nodes to reside in broadcast range, so single forwarding entries can map directly onto the wireless broadcast address. This approach shows the lowest memory demands, but only works well in small-scale single-hop deployments. A broadcast communication typically lacks corrective actions on the link-layer, *i.e.,* timeouts and retransmissions, and is generally discouraged in low-power networks with numerous network participants in the same wireless domain due to uncontrolled interferences.

6.2. Signaling overhead and handover delay assessment

We now theoretically inspect the incurred signaling overhead that results from device mobility by comparing relevant protocol features of HoPP, MAP-Me [20], MIPv6 [73], and its anchor-based multicast adaptation M-HMIPv6 [77]. To analyze handover effects and quantities, we assume a basic network topology where a single mobile node (MN) moves from a previous access router (PAR) to the next access router (NAR) as illustrated in Fig. 10. In case of (M-)HMIPv6, we consider the path stretch from PAR to NAR as shorter than to a home agent (HA). This simple setup rather intends to infer insights for the handover mechanisms than to develop a complete, but complex quantitative picture. Fig. 11 groups the signaling delays for the elemental protocol steps into three categories: (*i*) signaling that is geometry independent and occurs on a single hop, (*ii*) regional updates between the previous and current attachment points, and (*iii*) performing geometry dependent, global updates.

The selected protocols have different areas of application and therefore show varying implications on the link-layer. HoPP is designed for low-power multi-hop mesh networks without device associations on the link-layer. (M-)HMIPv6 and MAP-Me pursue general purpose deployments, which often follow star topologies when wirelessly connected. WiFi, as an example, requires device associations to an access point if not run in ad-hoc mode, which need to be re-established on device mobility. The layer 2 handoff time highly depends on the underlying link technology and hardware, but commonly approximates a delay of a few tens of milliseconds [77]. After successfully associating on the link-layer, an IPv6 node performs at minimum a local router discovery and an address configuration. The neighbor discovery protocol (NDP [78]) issues a link-local router solicitation (RS), which triggers a router advertisement (RA). Typically, these message exchanges yield a delay that is below 10 ms for most general link technologies with an exception for timeslotted solutions. HoPP manages its own prefix-specific routing system rooted at the Content Proxy (CP) and performs a similar task as soon as a node registers mobility: A scoped broadcast solicitation message (SOL) is transmitted to trigger a PAM from an upstream node. Similarly to RPL [68], a joining node inspects the ranks of multiple PAM responses to decide on the default router and then attaches to a particular DODAG branch. Here, delays similarly sum up to roughly 10–30 ms even with IEEE 802.15.4 as indicated in Section 5.3. MAP-Me does not specify the discovery process of default routers which suggests that it either relies on an orthogonal routing protocol to install the correct next-hop, or that it configures the hardware address of the associated peer as a next-hop.

After completing device re-associations and local configurations, (M-)HMIPv6 notifies the Mobility Anchor Point (MAP) about node mobility by sending a *Binding Update (BU)*. Respectively, MAP-Me signals the new location to the previous access router by sending an *Interest Update (IU)* to the previous location. In our scenario description, we consider device mobility within regional scope, *i.e.,* the new access router is only a few hops from the previous attachment point away. A delay in the range of a couple of tens of milliseconds – more in case of intermittent link failures, especially on wireless

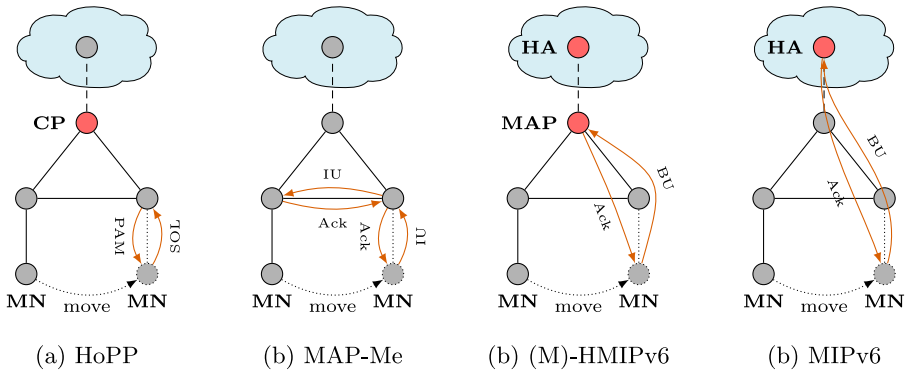


Fig. 10. Topology setup and mobility-related signaling for the selected protocol ensemble.

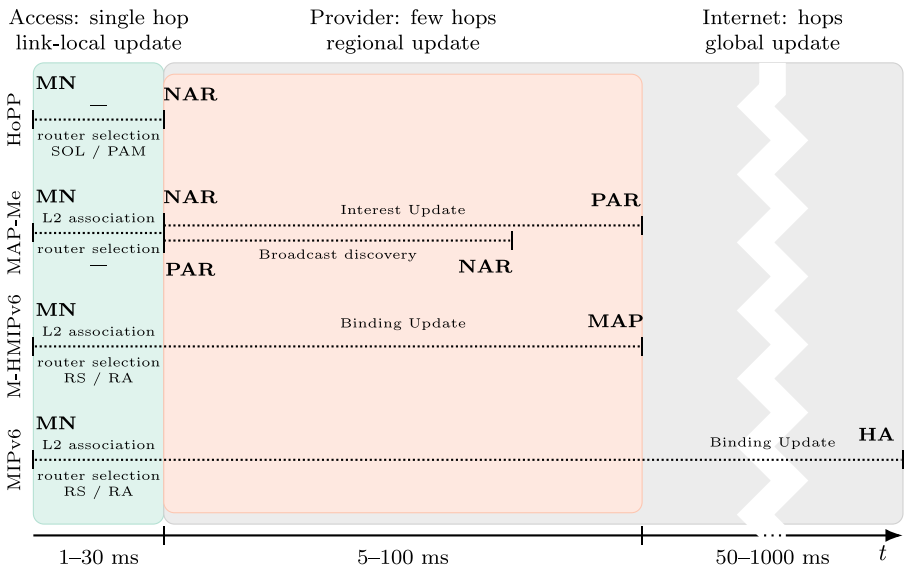


Fig. 11. Semi-quantitative comparison of handover signaling delays for a mobile node (MN) on device mobility from a previous (PAR) to next (NAR) access router using HoPP, MAP-Me, (M-)HMIPv6, and MIPv6. Signaling group into local, regional, and global updates.

media – is reasonable to assume. In contrast to (M-)HMIPv6 where BUs traverse end-to-end, the adaptive forwarding nature of NDN enables hop-wise state updates. While IUs propagate to the previous access router, any visited forwarder potentially installs the recent forwarding entry and can already divert ongoing traffic on path intersections to the new device location. Additionally, MAP-Me employs a quick discovery scheme to find breadcrumbs of disassociated mobile devices on neighboring attachment points by broadcasting ongoing traffic into the vicinity. If MNs leave breadcrumbs on attachment points in close range, then the consecutive broadcasting towards a specific MN can drastically decrease handover delays until a regional IU succeeds. On the other hand, defaulting to an unregulated broadcast is especially in dense wireless deployments harmful due to increased interferences and missing retransmission features on the link-layer.

MIPv6 does not employ similar shortcut methods to decrease the handover time, but rather relies on the geometry dependent binding update to the home agent. Since the path stretch between a new device location and the home agent may consist of an indefinite number of hops, the update procedure may require a period ranging from a few tens of milliseconds up to the order of seconds, in which a mobile node is unreachable. Conversely, HoPP does not require global routing updates for mobile devices since producers publish content per prefix to a content proxy that anchors the prefix in the underlying routing system. Subscribes retrieve data from content proxies, which decouples them from the actual producers. Data objects are hence still accessible even in the event of longer device sleep cycles or intermittent connectivity issues due to mobility.

In the following analytical evaluation of handover latencies, we assume the simple topology depicted in Fig. 10. For comparability reasons, the nodes in all protocol deployments connect in an ad-hoc fashion via IEEE 802.15.4, so we do not apply any association times on the link-layer when moving to a new access router. Table 3 summarizes the handover latency range for a single mobile node (MN) that moves from a previous access router to a new location. On a successful move, the MN performs a local router discovery and potential network configurations. Typically, this process requires at least one message round-trip (SOL-PAM, RS-RA) initiated by the MN. Former experimental testbed evaluations [2] indicate a single-hop round-trip time of roughly $10-20$ ms with similar radio configurations. In addition, MAP-Me and (M-)HMIPv6 perform a regional update that reaches two hops in our analytical model. Based on the previous round-trip assumptions, this adds another $20-40$ ms to the total handover latency. In the case of MIPv6, the Binding Updates are geometry dependent and traverse an indefinite number of hops, which can add a significant delay to the handover. Additionally, security considerations for (H)MIPv6 updates may require protective measures, *e.g.,* with IPsec, especially if they

Table 3
Handover latency ranges for mobility protocols based on the topological assessment in Fig. 10.

HoPP	MAP-Me	(M-)HMIPv6	MIPv6
10–20 ms	30–60 ms	30–60 ms	>60 ms

have global reach. While an increased packet overhead due to security features is insignificant on the Internet site, it is much more impactful on the IoT domain. Packets surpassing the maximum transmission unit (MTU) for IEEE 802.15.4 of 127 bytes get hop-wise fragmented by the 6LoWPAN [79] convergence layer and add further round-trips per hop for each fragment.

7. Conclusions and outlook

Node mobility and intermittent connectivity in low-power regimes severely challenge the routing between sensors and actuators. Long handover delays can result in extended downtime of nodes, or even partition a network topology. In this work, we found that (a) publish–subscribe with named topic prefixes can overcome the complexity of routing named data of things, and (b) NDN with *link-local* alerting has striking advantages for reactivity, security, and robustness in constrained environments.

We introduced the lightweight publish–subscribe system HoPP that was implemented in the CCN-lite network stack adaption of RIOT and experimentally evaluated in large, realistic testbeds with varying topologies. Our findings confirmed that the HoPP approach is robust and resilient while performing well in the majority of experiments. In particular, we could show that node mobility and temporary network partitioning require low repair overhead and can be quickly mitigated by local buffering and re-connects.

This work contributes insights and components for a future data-centric Web of Things. In future work, we will continue to investigate the different protocols and building blocks that can support the vision of a robust and secure web of constrained, loosely coupled things.

A note on reproducibility. We fully support reproducible research [80, 81] and perform all our experiments using open source software and an open access testbed. Code and documentation will be available on Github at https://github.com/inetrg/comnet-hopp-2021.

CRediT authorship contribution statement

Cenk Gündoğan: Conceptualization, Methodology, Software, Investigation, Visualization, Data curation, Writing – original draft. **Peter Kietzmann:** Conceptualization, Methodology, Validation. **Thomas C. Schmidt:** Conceptualization, Methodology, Writing – original draft, Writing – review & editing, Supervision, Project administration, Funding acquisition. **Matthias Wählisch:** Conceptualization, Methodology, Writing – original draft, Writing – review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.comnet.2021.108656.

References

[1] Q. Wang, J. Jiang, Comparative examination on architecture and protocol of industrial wireless sensor network standards, IEEE Commun. Surv. Tutor. 18 (3) (2016) 2197–2219.

[2] C. Gündogan, P. Kietzmann, M. Lenders, H. Petersen, T.C. Schmidt, M. Wählisch, 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, 2018, pp. 159–171, http://dx.doi.org/10.1145/3267955.3267967.

[3] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, B. Ohlman, A survey of information-centric networking, IEEE Commun. Mag. 50 (7) (2012) 26–36.

[4] G. Xylomenos, C.N. Ververidis, V.A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K.V. Katsaros, G.C. Polyzos, A survey of information-centric networking research, IEEE Commun. Surv. Tutor. 16 (2) (2014) 1024–1049.

[5] D. Lagutin, K. Visala, S. Tarkoma, Publish/subscribe for Internet: PSIRP perspective, Future Internet Assem. 84 (2010) 75–84.

[6] A. Carzaniga, M. Papalini, A.L. Wolf, Content-based publish/subscribe networking and information-centric networking, in: Proc. of the ACM SIGCOMM WS on Information-Centric Networking (ICN '11), ACM, New York, NY, USA, 2011, pp. 56–61.

[7] J. Chen, M. Arumaithurai, L. Jiao, X. Fu, K. Ramakrishnan, COPSS: An efficient content oriented Publish/Subscribe system, in: ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS'11), IEEE Computer Society, Los Alamitos, CA, USA, 2011, pp. 99–110.

[8] V. Jacobson, D.K. Smetters, J.D. Thornton, M.F. Plass, Networking named content, in: 5th Int. Conf. on Emerging Networking Experiments and Technologies (ACM CoNEXT'09), ACM, New York, NY, USA, 2009, pp. 1–12.

[9] C. Gündogan, P. Kietzmann, T.C. Schmidt, M. Wählisch, 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), IEEE Press, Piscataway, NJ, USA, 2018, pp. 331–334, http://dx.doi.org/10.1109/LCN.2018.8638030.

[10] X. Jiang, J. Taneja, J. Ortiz, A. Tavakoli, P. Dutta, J. Jeong, D. Culler, P. Levis, S. Shenker, An architecture for energy management in wireless sensor networks, SIGBED Rev. 4 (3) (2007) 31–36.

[11] J. Vasseur, Terms Used in Routing for Low-Power and Lossy Networks, RFC 7102, IETF, 2014.

[12] J. Hester, J. Sorber, The future of sensing is batteryless, intermittent, and awesome, in: Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems, in: SenSys '17, Association for Computing Machinery, New York, NY, USA, 2017, pp. 1–6.

[13] P. Ginzboorg, T. Kärkkäinen, A. Ruotsalainen, M. Andersson, J. Ott, DTN communication in a mine, in: 2nd Extreme Workshop on Communications, ACM, 2010.

[14] C. Gündogan, P. Kietzmann, T.C. Schmidt, M. Lenders, H. Petersen, M. Wählisch, M. Frey, F. Shzu-Juraschek, Information-centric Networking for the Industrial IoT, in: Proc. of 4th ACM Conference on Information-Centric Networking (ICN), Demo Session, ACM, New York, NY, USA, 2017, pp. 214–215.

[15] G. Tyson, N. Sastry, R. Cuevas, I. Rimac, A. Mauthe, A survey of mobility in information-centric networks, Commun. ACM 56 (12) (2013) 90–98.

[16] Y. Zhang, A. Afanasyev, J. Burke, L. Zhang, A survey of mobility support in named data networking, in: Proc. of IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), IEEE, Piscataway, NJ, USA, 2016, pp. 83–88.

[17] D. Kutscher, S. Eum, K. Pentikousis, I. Psaras, D. Corujo, D. Saucez, T. Schmidt, M. Waehlisch, Information-Centric Networking (ICN) Research Challenges, RFC 7927, IETF, 2016.

[18] Y. Zhang, Z. Xia, S. Mastorakis, L. Zhang, KITE: Producer mobility support in named data networking, in: Proceedings of the 5th ACM Conference on Information-Centric Networking, ACM, New York, NY, USA, 2018, pp. 125–136.

[19] D. Hernandez, L. Gameiro, C. Senna, M. Luís, S. Sargento, Handling producer and consumer mobility in IoT publish-subscribe named data networks, IEEE Internet Things J. (2021) Early access.

[20] J. Augé, G. Carofiglio, G. Grassi, L. Muscariello, G. Pau, X. Zeng, MAP-Me: Managing anchor-less producer mobility in content-centric networks, IEEE Trans. Netw. Serv. Manag. 15 (2) (2018) 596–610.

[21] D. Raychaudhuri, K. Nagaraja, A. Venkataramani, MobilityFirst: a robust and trustworthy mobility-centric architecture for the future internet, SIGMOBILE Mob. Comput. Commun. Rev. 16 (3) (2012) 2–13.

[22] F. Hermans, E. Ngai, P. Gunningberg, Global source mobility in the content-centric networking architecture, in: Proceedings of the 1st ACM Workshop on Emerging Name-Oriented Mobile Networking Design - Architecture, Algorithms, and Applications, in: NoM '12, ACM, New York, NY, USA, 2012, pp. 13–18.

[23] X. Jiang, J. Bi, Y. Wang, P. Lin, Z. Li, A content provider mobility solution of named data networking, in: Proc. of the 20th IEEE International Conference on Network Protocols (ICNP 2013), 2012, pp. 1–2.

[24] A. Azgin, R. Ravindran, A. Chakraborti, G. Wang, Seamless producer mobility as a service in information centric networks, in: Proceedings of the 3rd ACM Conference on Information-Centric Networking, in: ICN '16, ACM, New York, NY, USA, 2016, pp. 243–248.

[25] H.M. Islam, A. Lukyanenko, S. Tarkoma, A. Yla-Jaaski, Towards disruption tolerant ICN, in: 2015 IEEE Symposium on Computers and Communication (ISCC), IEEE, 2015, pp. 212–219.

[26] S. Arabi, E. Sabir, H. Elbiaze, Information-centric networking meets delay tolerant networking: Beyond edge caching, in: 2018 IEEE Wireless Communications and Networking Conference (WCNC), IEEE, 2018, pp. 1–6.

[27] G. Grassi, D. Pesavento, G. Pau, R. Vuyyuru, R. Wakikawa, L. Zhang, VANET via named data networking, in: 2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), IEEE, 2014, pp. 410–415.

[28] M. Meddeb, A. Dhraief, A. Belghith, T. Monteil, K. Drira, S. Gannouni, AFIRM: Adaptive forwarding based link recovery for mobility support in NDN/IoT networks, Future Gener. Comput. Syst. 87 (2018) 351–363.

[29] V. Sivaraman, D. Guha, B. Sikdar, Towards seamless producer mobility in information centric vehicular networks, in: 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring), IEEE, 2020, pp. 1–5.

[30] L. Wang, S. Bayhan, J. Ott, J. Kangasharju, A. Sathiaseelan, J. Crowcroft, Prodiluvian: Understanding scoped-flooding for content discovery in information-centric networking, in: 2Nd ACM Conference on Information-Centric Networking, in: ACM-ICN '15, ACM, New York, NY, USA, 2015, pp. 9–18.

[31] C. Yi, A. Afanasyev, I. Moiseenko, L. Wang, B. Zhang, L. Zhang, A Case for Stateful Forwarding Plane, Tech. Rep. NDN-0002, PARC, 2012.

[32] C. Yi, J. Abraham, A. Afanasyev, L. Wang, B. Zhang, L. Zhang, On the role of routing in named data networking, in: Proceedings of the 1st ACM Conference on Information-Centric Networking, in: ACM-ICN '14, ACM, New York, NY, USA, 2014, pp. 27–36.

[33] T.C. Schmidt, S. Wölke, N. Berg, M. Wählisch, Let's collect names: How PANINI limits FIB tables in name based routing, in: Proc. of 15th IFIP Networking Conference, IEEE Press, Piscataway, NJ, USA, 2016, pp. 458–466.

[34] J.J. Garcia-Luna-Aceves, M. Mirzazad-Barijough, A light-weight forwarding plane for content-centric networks, in: 2016 International Conference on Computing, Networking and Communications (ICNC), IEEE, 2016, pp. 1–7.

[35] B. Fenner, M. Handley, H. Holbrook, I. Kouvelas, Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised), RFC 4601, IETF, 2006.

[36] G. Xylomenos, X. Vasilakos, C. Tsilopoulos, V.A. Siris, G.C. Polyzos, Caching and mobility support in a publish–subscribe internet architecture, IEEE Commun. Mag. 50 (7) (2012) 52–58.

[37] U. Amozarrain, M. Larrea, Full Mobility and Fault Tolerance in Content-Based Publish/Subscribe, Tech. rep., University of the Basque Country UPV/EHU, 2019, URL http://www.sc.ehu.es/acwlaalm/research/EHU-KAT-IK-01-19.pdf.

[38] H. Jung, K. Choi, H. Kim, S. Kim, A networking scheme for large-scale pub/sub service over NDN, in: International Conference on Information and Communication Technology Convergence (ICTC '19), IEEE, 2019, pp. 1195–1200.

[39] W. Shang, A. Gawande, M. Zhang, A. Afanasyev, J. Burke, L. Wang, L. Zhang, Publish–subscribe communication in building management systems over named data networking, in: 2019 28th International Conference on Computer Communication and Networks (ICCCN), IEEE, 2019, pp. 1–10.

[40] T. Yu, Z. Zhang, X. Ma, P. Moll, L. Zhang, A Pub/Sub API for NDN-Lite with Built-in Security, Technical Report NDN-0071, University of California at Berkeley, 2021.

[41] J. Lee, S.M. Hwang, T. Abdelzaher, K. Marcus, K. Chan, Pub/sub-sum: A content summarization pub/sub protocol for information-centric networks, in: MILCOM 2019-2019 IEEE Military Communications Conference (MILCOM), 2019, pp. 847–852.

[42] A. Detti, D. Tassetto, N.B. Melazzi, F. Fedi, Exploiting content centric networking to develop topic-based, publish–subscribe MANET systems, Ad Hoc Netw. 24 (2015) 115–133.

[43] S.Y. Oh, D. Lau, M. Gerla, Content centric networking in tactical and emergency MANETs, in: 2010 IFIP Wireless Days, IEEE, Piscataway, NJ, USA, 2010, pp. 1–5.

[44] E. Baccelli, C. Mehlis, O. Hahm, T.C. Schmidt, M. Wählisch, Information centric networking in the IoT: Experiments with NDN in the Wild, in: Proc. of 1st ACM Conf. on Information-Centric Networking (ICN-2014), ACM, New York, 2014, pp. 77–86, http://dx.doi.org/10.1145/2660129.2660144.

[45] G.C. Polyzos, N. Fotiou, Building a reliable internet of things using information-centric networking, J. Reliab. Intell. Environ. 1 (1) (2015) 47–58.

[46] B. Mathieu, C. Westphal, P. Truong, Towards the usage of ccn for iot networks, in: Internet of Things (IoT) in 5G Mobile Technologies, Springer, Cham, Switzerland, 2016, pp. 3–24.

[47] J.J. Garcia-Luna-Aceves, ADN: An information-centric networking architecture for the Internet of Things, in: Proc. of the 2nd International Conference on Internet-of-Things Design and Implementation, in: IoTDI '17, ACM, New York, NY, USA, 2017, pp. 27–36.

[48] E.M. Schooler, D. Zage, J. Sedayao, H. Moustafa, A. Brown, M. Ambrosin, An architectural vision for a data-centric IoT: Rethinking Things, trust and clouds, in: IEEE 37th Intern. Conference on Distributed Computing Systems (ICDCS), IEEE, Piscataway, NJ, USA, 2017, pp. 1717–1728.

[49] C. Gündogan, J. Pfender, P. Kietzmann, T.C. Schmidt, M. Wählisch, On the impact of QoS management in an information-centric Internet of Things, Comput. Commun. 154 (2020) 160–172, http://dx.doi.org/10.1016/j.comcom.2020.02.046.

[50] M. Wählisch, T.C. Schmidt, M. Vahlenkamp, Bulk of interest: Performance measurement of content-centric routing, in: Proc. of ACM SIGCOMM, Poster Session, ACM, New York, 2012, pp. 99–100, URL http://conferences.sigcomm.org/sigcomm/2012/paper/sigcomm/p99.pdf.

[51] P. Gasti, G. Tsudik, E. Uzun, L. Zhang, DoS and DDoS in named data networking, in: Proc. of ICCCN, IEEE, 2013, pp. 1–7.

[52] M. Wählisch, T.C. Schmidt, M. Vahlenkamp, Backscatter from the data plane – threats to stability and security in information-centric network infrastructure, Comput. Netw. 57 (16) (2013) 3192–3206, http://dx.doi.org/10.1016/j.comnet.2013.07.009.

[53] S. Al-Sheikh, M. Wählisch, T.C. Schmidt, Revisiting countermeasures against NDN interest flooding, in: 2nd ACM Conference on Information-Centric Networking, Poster Session, ICN 2015, ACM, New York, 2015, pp. 195–196.

[54] O. Hahm, C. Adjih, E. Baccelli, T.C. Schmidt, M. Wählisch, ICN over TSCH: Potentials for link-layer adaptation in the IoT, in: Proc. of 3rd ACM Conf. on Information-Centric Networking (ICN 2016), Poster Session, ACM, New York, NY, USA, 2016, pp. 195—196, http://dx.doi.org/10.1145/2984356.2985226.

[55] C. Gündogan, P. Kietzmann, T.C. Schmidt, M. Wählisch, Designing a LoWPAN convergence layer for the information centric Internet of Things, Comput. Commun. 164 (1) (2020) 114–123, http://dx.doi.org/10.1016/j.comcom.2020.10.002.

[56] P. Kietzmann, C. Gündogan, T.C. Schmidt, O. Hahm, M. Wählisch, 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, 2017, pp. 36–42.

[57] J. Burke, P. Gasti, N. Nathan, G. Tsudik, Securing instrumented environments over content-centric networking: the case of lighting control and NDN, in: Computer Communications Workshops (INFOCOM WKSHPS), 2013 IEEE Conference on, IEEE, Piscataway, NJ, USA, 2013, pp. 394–398.

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

[59] R. Ravindran, A. Chakraborti, S. Amin, J. Chen, Support for Notifications in CCN, Internet-Draft – work in progress 01, IETF, 2017.

[60] M. Amadeo, C. Campolo, A. Iera, A. Molinaro, Information centric networking in IoT scenarios: The case of a smart home, in: Proc. of IEEE International Conference on Communications (ICC), IEEE, Piscataway, NJ, USA, 2015, pp. 648–653.

[61] D. Saxena, V. Raychoudhury, N. SriMahathi, SmartHealth-NDNoT: Named data network of things for healthcare services, in: Proc. of Workshop on Pervasive Wireless Healthcare (MobileHealth), ACM, New York, NY, USA, 2015, pp. 45–50.

[62] C. Tschudin, C. Scherb, et al., CCN lite: Lightweight implementation of the content centric networking protocol, 2018, URL http://ccn-lite.net.

[63] E. Baccelli, C. Gündogan, O. Hahm, P. Kietzmann, M. Lenders, H. Petersen, K. Schleiser, T.C. Schmidt, M. Wählisch, Riot: an open source operating system for low-end embedded devices in the IoT, IEEE Internet Things J. 5 (6) (2018) 4428–4440, http://dx.doi.org/10.1109/JIOT.2018.2815038.

[64] B. Ahlgren, A. Lindgren, Y. Wu, Demo: Experimental feasibility study of CCN-lite on contiki motes for IoT data streams, in: Proceedings of the 2016 Conference on 3rd ACM Conference on Information-Centric Networking, ACM, New York, NY, USA, 2016, pp. 221–222.

[65] W. Shang, A. Afanasyev, L. Zhang, The design and implementation of the NDN protocol stack for RIOT-OS, in: Proc. of IEEE GLOBECOM 2016, IEEE, Washington, DC, USA, 2016, pp. 1–6.

[66] C. Gündogan, T.C. Schmidt, M. Wählisch, C. Scherb, C. Marxer, C. Tschudin, ICN Adaptation To LowPAN Networks (ICN LoWPAN), IRTF Internet Draft – work in progress 10, IRTF, 2021, URL https://tools.ietf.org/html/draft-irtf-icnrg-icnlowpan.

[67] C. Gündogan, P. Kietzmann, T.C. Schmidt, M. Lenders, H. Petersen, M. Wählisch, M. Frey, F. Shzu-Juraschek, Demo: Seamless producer mobility for the industrial information-centric internet, in: Proc. of 16th ACM International Conference on Mobile Systems, Applications (MobiSys), Demo Session, ACM, New York, NY, USA, 2018.

[68] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, R. Alexander, RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks, RFC 6550, IETF, 2012.

[69] P. Levis, T. Clausen, J. Hui, O. Gnawali, J. Ko, The Trickle Algorithm, RFC 6206, IETF, 2011.

[70] O. Gnawali, P. Levis, The Minimum Rank with Hysteresis Objective Function, RFC 6719, IETF, 2012.

[71] Z. Shelby, S. Chakrabarti, E. Nordmark, C. Bormann, Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs), RFC 6775, IETF, 2012.

[72] C. Tschudin, C. Wood, M. Mosko, D. Oran, File-Like ICN Collections (FLIC), Internet-Draft – work in progress 02, IETF, 2019.

[73] C. Perkins, D. Johnson, J. Arkko, Mobility Support in IPv6, RFC 6275, IETF, 2011.

[74] C. Adjih, E. Baccelli, E. Fleury, G. Harter, N. Mitton, T. Noel, R. Pissard-Gibollet, F. Saint-Marcel, G. Schreiner, J. Vandaele, T. Watteyne, FIT IoT-LAB: A large scale open experimental IoT testbed, in: 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT), 2015, pp. 459–464.
[75] C. Bormann, M. Ersue, A. Keranen, Terminology for Constrained-Node Networks, RFC 7228, IETF, 2014.
[76] M. Hoque, S.O. Amin, A. Alyyan, B. Zhang, L. Zhang, L. Wang, NLSR: Named-data link state routing protocol, in: 3rd ACM SIGCOMM Workshop on Information-Centric Networking, in: ICN '13, ACM, New York, NY, USA, 2013, pp. 15–20.
[77] T.C. Schmidt, M. Wählisch, Predictive versus reactive – analysis of handover performance and its implications on IPv6 and multicast mobility, Telecommun. Syst. 30 (1–3) (2005) 123–142, http://dx.doi.org/10.1007/s11235-005-4321-4.
[78] T. Narten, E. Nordmark, W. Simpson, H. Soliman, Neighbor Discovery for IP Version 6 (IPv6), RFC 4861, IETF, 2007.
[79] G. Montenegro, N. Kushalnagar, J. Hui, D. Culler, Transmission of IPv6 Packets over IEEE 802.15.4 Networks, RFC 4944, IETF, 2007.
[80] ACM, Result and artifact review and badging, 2017, http://acm.org/publications/policies/artifact-review-badging.
[81] Q. Scheitle, M. Wählisch, O. Gasser, T.C. Schmidt, G. Carle, Towards an ecosystem for reproducible research in computer networking, in: Proc. of ACM SIGCOMM Reproducibility Workshop, ACM, New York, NY, USA, 2017, pp. 5–8.

Cenk Gündogăn received the M.Sc. degree in computer science from the Institut für Informatik, Freie Universität Berlin, Germany, in 2016. Currently, he is pursuing the Ph.D. degree with the Internet Technologies Group, Hamburg University of Applied Sciences, Germany, and explored within the I3 project—Information Centric Networking (ICN) for the Industrial Internet—routing, QoS, and resilience in ICN-based and IoT tailored networks. Recently, Cenk Gündogăn put focus on a data-centric Web of Things deployment option by applying ICN principles to the IETF envisioned IoT network stack. He is one of the core developers and maintainer of RIOT.

Peter Kietzmann received the M.Eng. degree in information technology from the Hamburg University of Applied Sciences, Hamburg, Germany, where he is currently pursuing the Ph.D. degree with the Internet Technologies Research Group. His particular research interest includes low-power radios, and IoT protocols, many of which he analyzed and transformed into code of RIOT. In the German research project I3 (ICN for the Industrial Internet) he explores IoT-based technologies for information centric networks and security components.

Thomas C. Schmidt is professor of Computer Networks and Internet Technologies at Hamburg University of Applied Sciences (HAW), where he heads the Internet Technologies research group (iNET). Prior to moving to Hamburg, he was director of a scientific computer centre in Berlin. He studied mathematics, physics and German literature at Freie Universitaet Berlin and University of Maryland, and received his Ph.D. from FU Berlin in 1993. Since then he has continuously conducted numerous national and international research projects. He was the principal investigator in a number of EU, nationally funded and industrial projects as well as visiting professor at the University of Reading, U.K. His continued interests lie in the development, measurement, and analysis of large-scale distributed systems like the Internet. He serves as co-editor and technical expert in many occasions and is actively involved in the work of IETF and IRTF. Together with his group he pioneered work on an information-centric Industrial IoT and the emerging data-centric Web of Things. Thomas is a co-founder of several large open source projects and coordinator of the community developing the RIOT operating system—the friendly OS for the Internet of Things.

Matthias Wählisch is an Assistant Professor of Computer Science at Freie Universität Berlin where he heads the Internet Technologies Research Lab. He received his Ph.D. in computer science with highest honors from Freie Universität Berlin. His research and teaching focus on efficient, reliable, and secure Internet communication. This includes the design and evaluation of networking protocols and architectures, as well as Internet measurements and analysis. His efforts are driven by improving Internet communication based on sound research. Matthias is the PI of several national and international projects, supported by overall 4.7M EUR grant money. He published more than 150 peer-reviewed papers (e.g., at ACM HotNets, ACM IMC, The Web Conference). Since 2005, Matthias is active within IETF/IRTF, including eight RFCs and several Internet drafts. His research results have been distinguished multiple times. Amongst others, he received the Young Talents Award of Leibniz-Kolleg Potsdam for outstanding achievements in advancing the Internet, as well as the Excellent Young Scientists Award (10,000 EUR) for his contributions to the Internet of Things and their prospective entrepreneurial practice. He co-founded some successful open source projects such as RIOT, where he is still responsible for the strategic development.