NDN, CoAP, and MQTT: A Comparative Measurement Study in the IoT

Cenk Gündoğan
HAW Hamburg
cenk.guendogan@haw-hamburg.de

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

Martine Lenders
Freie Universität Berlin
m.lenders@fu-berlin.de

Hauke Petersen
Freie Universität Berlin
hauke.petersen@fu-berlin.de

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

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

ABSTRACT

This paper takes a comprehensive view on the protocol stacks that are under debate for a future Internet of Things (IoT). It addresses the holistic question of which solution is beneficial for common IoT use cases. We deploy NDN and the two popular IP-based application protocols, CoAP and MQTT, in its different variants on a large-scale IoT testbed in single- and multi-hop scenarios. We analyze the use cases of scheduled periodic and unscheduled traffic under varying loads. Our findings indicate that (a) NDN admits the most resource-friendly deployment on nodes, and (b) shows superior robustness and resilience in multi-hop scenarios, while (c) the IP protocols operate at less overhead and higher speed in single-hop deployments. Most strikingly we find that NDN-based protocols are in significantly better flow balance than the UDP-based IP protocols and require fewer corrective actions.

CCS CONCEPTS

• **Networks** → **Network protocol design**; **Network performance analysis**; **Network experimentation**;

KEYWORDS

Internet of Things; wireless; security; energy; measurement; protocol evaluation

ACM Reference Format:
Cenk Gündoğan, 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 *ICN '18: 5th ACM Conference on Information-Centric Networking (ICN '18), September 21–23, 2018, Boston, MA, USA.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3267955.3267967

ICN '18, September 21–23, 2018, Boston, MA, USA

ACM ISBN 978-1-4503-5959-7/18/09…$15.00
https://doi.org/10.1145/3267955.3267967

1 INTRODUCTION

The Internet of Things (IoT) is evolving and an increasing number of controllers in the field is augmented with network interfaces that speak IP. Current deployments often are part of larger systems (e.g., a heating) or attached to infrastructure (e.g., smart city lighting). Such devices connect to power, use common broadband links, and adopt the old MQTT protocol [11] for publishing IoT data to a remote cloud. The prevalent use case forecasted for the IoT, though, consists of billions of constrained sensors and actuators mainly not cabled to power, but connected via low power lossy wireless links. The key target of the IoT will be data, of which a total of 1.6 Zettabytes is expected in 2020 [40]. The mass constituents of the IoT will be tiny, cheap *things* that are severely challenged by the current way of connecting to the Internet.

This new class of connected devices cannot be integrated into today's Internet infrastructure without technologies that bridge the scale. The IETF has designed a suite of protocols for successfully serving the needs of a constrained IoT. IPv6 adaptation layers such as 6LoWPAN [42] enable a deployment on constraint links (e.g., IEEE 802.15.4), which RPL routing arranges in a multi-hop topology [62]. The Constrained Application Protocol (CoAP) [53] offers a lightweight alternative to HTTP while running over UDP, or DTLS [45] for session security. This set of solutions extends the host-centric end-to-end paradigm of the Internet to the embedded world and puts IPv6 in place for loosely joining the *things*.

However, doubts arose whether host-to-host sessions are the appropriate approach in these disruption-prone environments of (wireless) things, and the data-centric nature at the Internet edge called for rethinking the current IoT architecture [49]. ICN networks [2] have been identified as promising candidates to replace the rather complex IETF network stack in a future IoT. Name-based routing and in-network caching as contributed by Named Data Networking (NDN) [31, 63] bear the potential to increase robustness of application scenarios in regimes of low reliability and reduced infrastructure (e.g., without DNS). Following initial concepts [43] and early experimental work [10], the adaptation, analysis, and deployment of NDN for the IoT became an active research area that advocated the IoT as a candidate of early NDN adoption.

Still open problems persist, namely naming, routing, forwarding [61], and data push [36] as Shang et al. [51] recently reminded.

While the IETF and the ICN community tweak their protocols and companies deploy MQTT-to-cloud uplinks, the quest for the best solution remains open. Rather little is known about the differences and commonalities when deploying the varying approaches in the wild. This surprisingly unsatisfying state of the art motivates us to implement, deploy, and thoroughly analyze the different protocols in typical use cases and scenarios for the constrained Internet of Things.

The main contribution of this paper is a thorough comparative analysis of the three protocol families NDN, CoAP, and MQTT[1] covering its main variants. We implemented characteristic IoT use cases for ten variants of these protocols, deployed them in single- and multi-hop scenarios on a large IoT testbed, and ran competitive performance contests under fully equivalent conditions. Our analysis showed common behavior for pull versus push solutions in single-hop experiments, but revealed significant differences in the challenging multi-hop domain. Flow performance, reliability and stability attained superior results for hop-wise replicated NDN protocols, while end-to-end approaches showed severe shortcomings at iterated link stress.

The remainder of this paper is structured as follows. The following Section 2 summarizes the key protocol properties and introduces the use cases. Section 3 explains our implementations and experimental setup. We present measurements and analyze the results in Section 4, In Section 5, we revisit the related work and conclude in Section 6.

2 BACKGROUND AND USE CASES

2.1 CoAP

CoAP, the Constrained Application Protocol [53], was designed to support REST services in machine to machine communication. Basically, it aims for replacing HTTP on constrained nodes. In contrast to HTTP, CoAP is able to run on top of UDP and introduces a lean transactional messaging layer to compensate for the connectionless transport. CoAP provides a more compact header structure than HTTP.

Three communication primitives are currently supported by this extensible protocol: (*i*) pull, (*ii*) push, and (*iii*) observe. Pull implements the common request response communication pattern. However, as IoT scenarios also include the pro-active communication of unscheduled state changes, CoAP was extended to support pushing new events to its peers. Still, this does not allow for publish-subscribe scenarios when producer and consumer are decoupled in time and data is not yet available at the request. The support for delayed data delivery in publish-subscribe was specified in CoAP observe [28]. Here, clients can signal interest in observing data, which basically means that a CoAP server delivers data as soon as available and maintains state until clients explicitly unsubscribe.

[1]We use the UDP-adapted version MQTT-SN, since TCP is inappropriate for the constrained IoT.

CoAP must be considered as the IETF standard to implement application layer data transfer in the future Internet of Things. Currently, several implementations exist, as well as early adoption in a few selected products and deployments.

2.2 MQTT

MQTT [11], the Message Queue Telemetry Transport, was designed as a publish-subscribe messaging protocol between clients and brokers. Clients can publish content, subscribe to content, or both. Servers (commonly called *broker*) distribute messages between publishing and subscribing clients. It is worth noting that the protocol is symmetric: Clients as well as brokers can be sender and receiver when MQTT delivers application messages.

MQTT is considered a lightweight protocol for two reasons. First, it provides a lean header structure, which reduces packet parsing and makes it suitable for IoT devices with low energy resources. Second, it is easy to implement. In its simplest form, MQTT offloads reliability support completely onto TCP.

To provide flexible Quality of Service on top of the underlying transport, MQTT defines three QoS levels, which reflect the agreement regarding message transfer between broker and consumer – both can be sender and receiver. *QoS 0* implements unacknowledged data transfer. An MQTT receiver gets a message at most once, depending on the capabilities of the underlying network, as there is no retransmission on the application layer. *QoS 1* guarantees that a message is delivered at least once. This requires that a message is stored at the sender side until an acknowledgement was received. Based on timeouts, an MQTT sender will retransmit application messages when an acknowledgement is missing. *QoS 2* ensures that a message is received exactly once, to avoid packet loss or processing of duplicates at the MQTT receiver side. This requires a two-step acknowledgement process and more states at both sides.

To adapt MQTT to constrained networks which are based on low data rates and very small packet lengths such as in 802.15.4, MQTT-SN [55] is specified. Header complexity is reduced by replacing topic strings by topic IDs, to identify content. In contrast to MQTT, MQTT-SN is able to run on top of UDP. It still supports all QoS levels but does not inherit any reliability property from the transport layer.

2.3 ICN Protocols

The core **NDN** protocol [31, 63] combines name-based routing from TRIAD [22] and stateful forwarding from DONA [34] to implement a request response scheme on the network layer. Any consumer can request data that is subsequently delivered along a trail of reverse path forwarding states. As an important feature, data will only be delivered to those who requested the data. This means that data must be (individually) named at the Interest request and that yet unavailable data requires repeating Interests until the application receives the data.

Table 1: Comparison of CoAP, MQTT, and ICN protocols. CoAP and MQTT support reliability only in confirmable mode (c) and QoS levels 1 and 2 (Q1, Q2).

	Current IoT Protocols					ICN Protocols		
	CoAP [53]			MQTT [11]	MQTT-SN [55]	NDN [31, 63]	I-Not [4]	HoPP [24]
	PUT	GET	Observe					
Transport	UDP	UDP	UDP	TCP	UDP	n/a	n/a	n/a
Pub/Sub	✘	✘	✓	✓	✓	✘	✘	✓
Push	✓	✘	✓	✓	✓	✘	✓	✘
Pull	✘	✓	✘	✘	✘	✓	✘	✓
Flow Control	✘	✘	✘	✓	✘	✓	✘	✓
Reliability	(c)	(c)	✘	(Q1, Q2)	(Q1, Q2)	✓	✓	✓

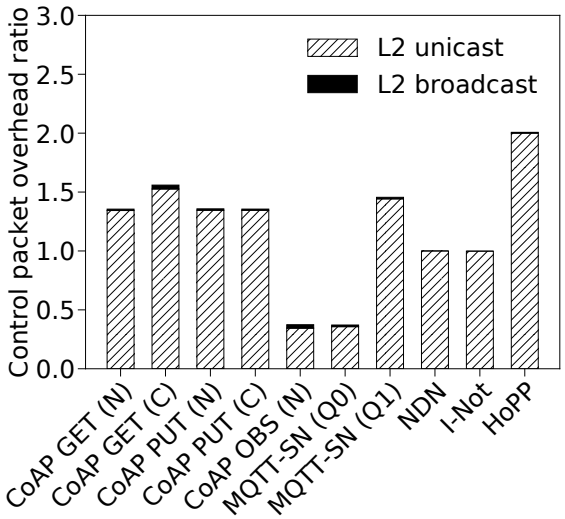


Figure 1: Relative protocol overhead under relaxed network conditions incl. topology control broadcasts.

The lack of push primitives in NDN triggered the idea of inverting the NDN semantic by placing data in an **Interest Notification (I-Not)** which in turn gets acknowledged by the subsequent (empty) data packet. This idea was originally proposed in [4] and was since then criticized for its lack of (*i*) caching support, (*ii*) flow control, and (*iii*) DDoS resilience.

Several publish-subscribe extensions have been proposed for NDN (COPSS [15], PSync [64]) to provide further decoupling of consumers and data sources. As COPSS relies on a persistent forwarding infrastructure and PSync on Interest broadcasting, both schemes do not satisfy the requirements of the constrained IoT. Our lightweight IoT variant **HoP and Pull (HoPP)** [24] provides a publish-subscribe system for constrained IoT deployments based on ICN/NDN principles. A constrained IoT publisher announces a name towards a content proxy to trigger content requests and to replicate the data towards a content proxy (or broker). Forwarding nodes on the path between publisher and content proxy hop-wise request content for this name by using common Interest and data messages. A content subscriber in HoPP behaves almost like any content requester in NDN and issues a regular Interest request towards the content proxy CP. However, in contrast to NDN (*i*) a subscriber cannot extract content names from its FIB, since FIBs only contain PANINI default routes [48], but uses application-specific topic tables instead; (*ii*) it does not expect an immediate reply, but issues Interests with extended lifetimes. HoPP enables rapid communication of unscheduled data events. It operates at a similar timescale as push protocols without actually pushing data.

2.4 Protocol Comparison and Use Cases

Key properties of the three protocol families NDN, CoAP, and MQTT and its variants are compared in Table 1. Specialized properties of the different approaches become apparent: Every protocol variant features distinct capabilities. Notably in the IoT, where TCP (aka generic MQTT) is unavailable, the pull-based NDN and NDN-HoPP are the only protocols admitting flow control and reliability as a generic service.

An additional mechanism for link recovery and retransmission has been brought to NDN with NDNLP [54]. Facing the lossy nature of low-power wireless links in the IoT, it may be tempting to deploy this additional protocol to enhance the overall reliability. However, common radio links like IEEE 802.15.4 already feature ARQs (Automatic Repeat-reQuests), and a network layer link would put a second acknowledgement to the air, which in turn would increase the omnipresent risk of interference. For this reason, we did neither deploy nor further investigate NDNLP in our further analyses.

Figure 1 compares the control overhead for all protocol variants under consideration as obtained from experiments under relaxed network conditions at negligible interference. Aside from topology building and maintenance that are mainly broadcasts (marked in Fig. 1), common request protocols require one request per data item, whereas publish-subscribe schemes only require subscription notification per topic. As a pull protocol, HoPP requires requests and an additional message to advertise names.

Common IoT deployment use cases consist of stub networks as visualized in Figure 2 that may be single- or multi-hop. Traffic flows from or to the IoT edge nodes in three patterns: (*i*) scheduled periodic sensor readings, (*ii*) unscheduled and

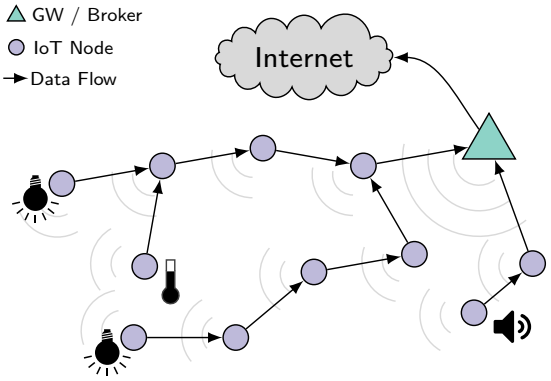


Figure 2: Use case scenario of a multi-hop IoT topology.

uncoordinated data updates, or (*iii*) on demand notifications or alerting. It is worth noting that the different protocol properties (e.g., push versus pull versus pub-sub) can serve these alternating needs in a quite distinct manner.

3 IMPLEMENTATION AND EXPERIMENTAL SETUP

Software Platforms. On the IoT nodes, all of our experiments are based on RIOT version `2018.01`. To analyze CoAP, MQTT-SN, and NDN we use `gCoAP`, `Asymcute`, and `CCN-lite` respectively. All three protocol implementations are part of the common RIOT release and thus reflect typical software components used in low-end IoT scenarios.

On the brokers or gateways, the testbed infrastructure deploys Linux systems. To support MQTT broker and CoAP observe client as well as CoAP PUT server functionalities, we used `aiocoap` version 0.3 and `mosquitto.rsmb` version 1.3.0.2. Both are popular open source implementations in this context.

Testbed. We conduct our experiments in the FIT IoT-LAB testbed. The hardware platform consists of typical class 2 devices [13] and features an ARM Cortex-M3 MCU with 64 kB of RAM and 512 kB of ROM. Each device is equipped with an Atmel AT86RF231 [7] transceiver to operate on the IEEE 802.15.4 radio. The gateway runs on a Cortex-A8 node, which is more powerful than the M3 edge nodes.

The testbed provides access to several sites with varying properties. We perform our experiments on two sites, to analyze single-hop as well as multi-hop scenarios.

Single-hop topology The *Paris* site consists of approximately 70 nodes, which are within the same radio range. We choose two arbitrary nodes and run all single-hop experiments on them. One node is a content producer, the other node acts as consumer (gateway/broker).

Multi-hop topology The *Grenoble* site consists of approximately 350 nodes spread evenly in the Inria Grenoble building. We choose 50 M3 nodes (low-end IoT device) and one A8 node (gateway/broker) arbitrarily and run all multi-hop experiments on them. All low-end devices operate as content producers. In our CoAP and MQTT experiments, we use RPL to build and maintain the routing topology across all nodes. In our NDN-based experiments, we build tree topologies analogously as HoPP does. In any case, we ensure that all protocols use the same routing topology for comparison. Typical path length are four to five hops.

Scenarios and Parameters. We align all experiments with respect to the configurations of retransmissions and timeouts to ensure comparability among protocols. All protocols employ the same retransmission strategy: In case of failures, each node waits 2 seconds before retransmitting the original application or control data. For NDN, HoPP and I-Not, retransmissions are performed hop-by-hop, while CoAP and MQTT perform them end-to-end. At most 4 retransmissions will occur for each data. Interest lifetimes are configured to 10 seconds for NDN based protocols to limit PIT memory consumption. We repeat each experiment 1,000 times.

To accommodate all 50 nodes in the routing topology, the FIB size was adjusted accordingly on each IoT node. For CoAP and MQTT, this translates in our IPv6 scenario to a FIB size of 50 entries with roughly 32 bytes each (`sizeof(destination)` + `sizeof(next-hop)`). In our NDN scenarios, each node owns a unique prefix of the form $/\rho_i$ with a length of `24` bytes. The next-hop face of each FIB entry points to the `8`-byte IEEE 802.15.4 link-layer address. In total, this setup yields comparable size requirements for all scenarios.

In the NDN scenarios, we use unique content names prefixed by $/\rho_i$ with incremental local packet counters. CoAP works without unique names but uses common URIs. The MQTT-SN protocols register a common topic name, similar to CoAP, and publish under a unique topic ID thereafter. In all scenarios, the data is of the same JSON format consisting of a unique identifier and a sensor value attribute. These short messages can be accommodated by the link layer and do not require fragmentation. It is noteworthy that we neither applied header compression in the IP [12] nor in the NDN world [25].

4 EVALUATION AND RESULTS

4.1 Analyses and Metrics

The objective of this work is to quantify the efficiency and utility of the considered protocols in real deployment scenarios. With this in mind, we want to shed light on resource consumption and the operational properties of data dissemination from different angles and in the different deployment use cases.

In detail, we analyze the *memory consumption* on nodes, the effective *network utilization* by control and data traffic including *protocol overhead* and *link stress* caused by retransmissions. The actual performance of data transmission is measured in *data loss*, *goodput*, and *content arrival time* which represents the delay between issuing a transaction and data arrival at the sink. Here, we use the term *time to completion* interchangeably. We also consider the *data flows*

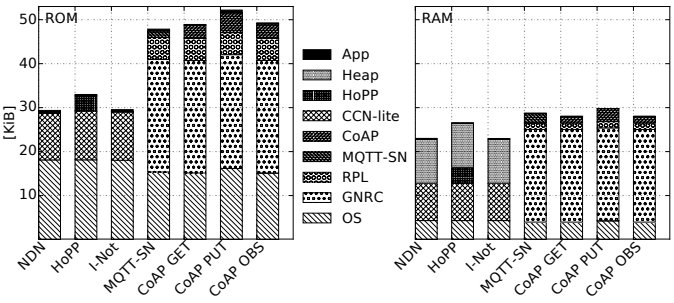


Figure 3: Resource consumption of ROM (left) and RAM (right) for the different software stacks.

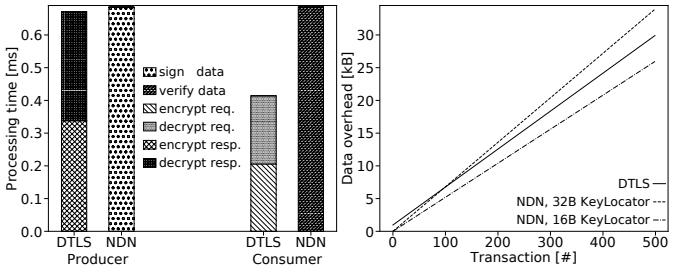


Figure 4: Security overheads—CPU consumption (left) and data overhead (right) per content transaction for IP/DTLS and NDN/HMAC.

and its *energy consumption*. These multi-sided analyses are performed on complete packet traces which we recorded from the different experiments, and a monitoring of the system state at participating nodes.

Security measures largely differ between the IP and the ICN world. DTLS [45] provides privacy and integrity for UDP datagrams within sessions based on pre-established private keys. NDN authenticates data chunks between arbitrary endpoints without the need for session state. Canonically, asymmetric signatures are attached to data chunks in NDN, but since the complexity of asymmetric crypto exceeds the capabilities of constrained nodes, keyed-hash message authentication code (HMAC) can also be applied. The use of HMAC likewise relies on pre-established keys.

In both worlds, security extensions add message and processing overhead, but do not change the overall behavior of the protocols. For this reason, we compare security overheads in separate micro-benchmarks and perform the remaining experiments without applying the corresponding security measures.

We do not consider network congestion from external cross-traffic in this work. However, each individual transmission experiences self-induced background traffic from the experiment that differs for varying request/publish intervals and jitter. On average, this side-traffic is constant per experimental run.

4.2 Protocol Stack Sizes

Largely differing properties and complexities of the protocol variants under test lead to seven distinct software stacks. Nodal memory consumption for these different protocol stacks are depicted in Figure 3. We differentiate the protocol layers in place to disclose the details.

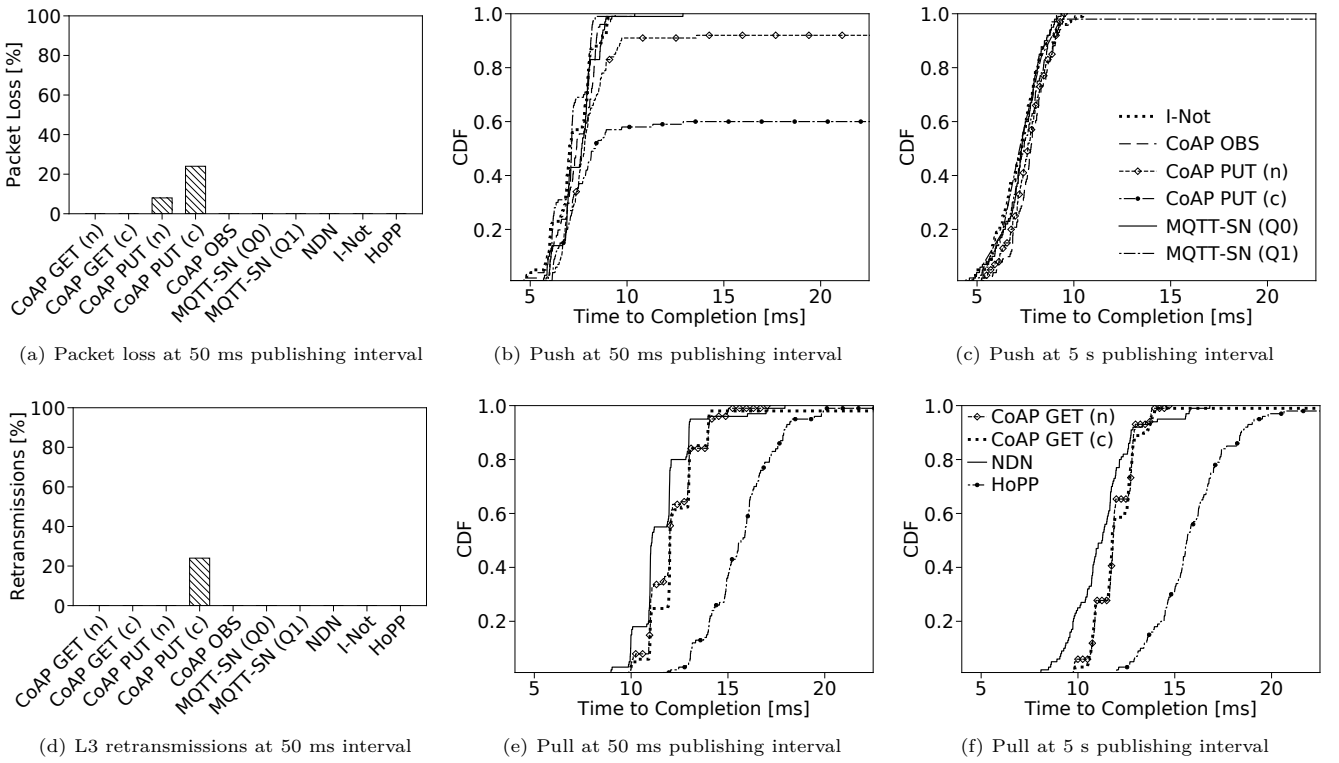
Main memory is the scarcest resource in the IoT. While protocols require OS support of 4,060 B (MQTT-SN) – 4,400 B (NDN) kernels, NDN admits the leanest stack of 8,700 B consumed by CCN-lite. All IP protocol stacks are significantly larger and approximately triple the size of CCN-lite. On the overall, about 30 KiB are needed to host IP protocols, leaving only a few dozen KiBs for the application on typical constrained nodes. All ICN protocols provide a Content Store (CS) of 10,240 B on the heap, which is the price of in-network caching. It should be noted that the GNRC network stack contributes a packet buffer to both, the IP and the ICN world that is also used for retransmissions [37]. Program sizes of NDN protocols are much smaller and consume about 40 % less ROM. The operating system support varies with protocol requirements on the highly modular RIOT OS platform.

4.3 Security Overheads

Many use cases of the IoT rely on integrity and authenticity of the collected data. Security extensions of the communication protocols are requested to ensure those properties at costs which we are now evaluating. For our micro-benchmarks of the IP world, we fixed the scenario of a DTLS session established between two nodes. We quantify the messaging overhead obtained from a single session establishment and the packet overhead as a function of data transactions—the request/response-guided transfer of a data unit. We also recorded the CPU expenses at the content producer and consumer per transaction.

The most comparable scenario for NDN consists of HMAC-based authentication of data using SHA256 per chunk. For quantifying the overheads in data packets, we chose two common sizes of the KeyLocatorTLV: 16B and 32B.

Figure 4 visualizes the results of our security benchmarks performed on the IoT-LAB M3 nodes. While message overheads for NDN are similar or better (for 16B KeyLocatorTLV), DTLS data verification can be performed at two-thirds of the NDN costs. It should be noted, though, that the different security models of DTLS and NDN make comparisons difficult. While DTLS operates within an established session that is strictly bound to endpoints, the content of NDN can be replicated between varying nodes. In particular, the NDN approach is robust under mobility and network changes, whereas DTLS would require to re-establish sessions in many cases at significant cost. Conversely, only DTLS encrypts transport payloads and thereby contributes data privacy.



(a) Packet loss at 50 ms publishing interval

(b) Push at 50 ms publishing interval

(c) Push at 5 s publishing interval

(d) L3 retransmissions at 50 ms interval

(e) Pull at 50 ms publishing interval

(f) Pull at 5 s publishing interval

Figure 5: Time to content arrival for scheduled publishing in a single-hop topology at different intervals.

4.4 Single-Hop with Scheduled Publishing

Protocol performances are first evaluated in a single-hop topology at the Paris testbed with periodic content publishing every 50 *ms* and 5 *s*. Content is pushed or requested accordingly. Figure 5 displays the results for protocol reliability and temporal performance. As an overall trend, it is apparent that push-oriented protocols operate faster, but less reliable.

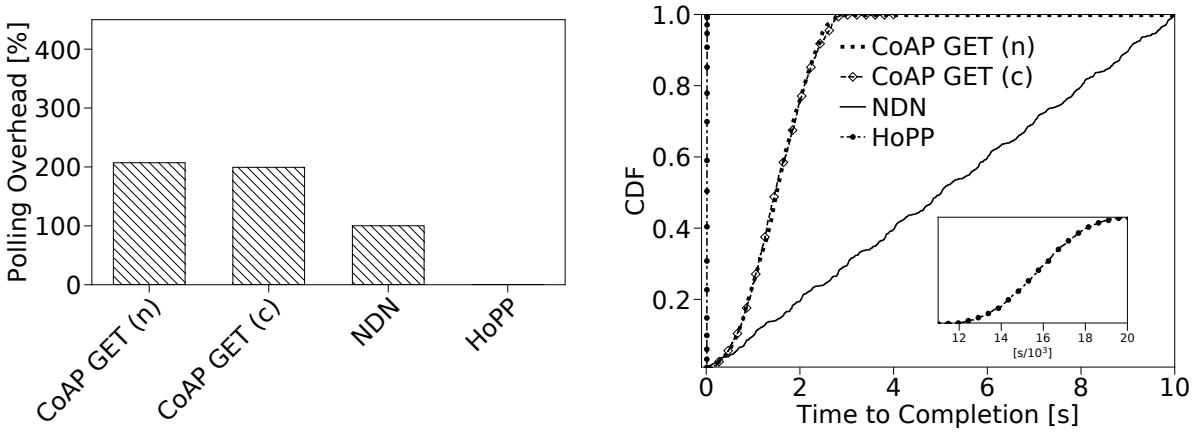
For the rather relaxed scenario of publishing every 5 *s*, we see the protocol families in rough agreement. Push-based protocols require an average of 7 *ms* (Fig. 5(c)) for data delivery, pull-based protocols take 11 *ms* (Fig. 5(f)), with the exception of HoPP which is slightly slower on this short timescale due to its three-way handshake.

The publishing interval of 50 *ms* puts some protocols under stress, even though IEEE 802.15.4 practically limits transmission only below an interval of 10 *ms*. The performance of CoAP PUT significantly degrades (Fig. 5(b)), leaving the unconfirmed messaging at a total data loss of 6 % (Fig. 5(a)). The PUT of Confirmable CoAP instead initiates 26 % retransmissions (Fig. 5(d)) which increase delays up to a factor of five. Confirmable CoAP does complete data delivery at 42 *ms* (Fig. 5(b) is clipped for visibility). It should be noted, though, that retransmissions on the data link layer are present for all protocols and are reflected by the staircase patterns. We do not measure these fast repeats ($\leq$ 10 *ms*) in this work, but refer to our previous study [33] for further details.

4.5 Single-Hop with Unscheduled Publishing

Our next experiments address the common IoT use case of publishing data at irregular intervals. This is the typical pattern for observing third party actions (*e.g.,* light switching), or largely uncoordinated sensing environments. Push-based protocols naturally serve these application needs. We quantify the behavior of the request-based protocols in practice and chose the moderate setting of publishing content every two seconds on average. Publishing is uniformly distributed in the interval of [1 s ... 3 s]. The protocols CoAP and NDN request the content periodically every second so that updates are not lost.

Figure 6(b) visualizes content delivery times for all request-oriented protocols. CoAP GET and NDN now operate on a timescale of seconds, while HoPP continues to complete in the unaltered range of 15 *ms* without additional protocol operations – the unsurprising outcome of content triggers built into HoPP. CoAP requests content using a common name with the result of likely duplicate content transmissions. On average, CoAP needs two requests to retrieve fresh content with the expected average delay of $\approx$ 2 *s* and a corresponding polling overhead of 200 % (Fig. 6). In contrast, NDN admits



(a) Control Overhead for polling unscheduled content

(b) Time to unscheduled content arrival

Figure 6: Pull protocol performance at random publishing in [1s ... 3s].

lower overhead, as Interests are locally managed at the PIT and only retransmitted after state timeout.

However, issuing Interests at a higher rate than content arrival leads to an accumulation of open states in the PIT. As resources on the constrained nodes are tightly bound, the PIT limits are quickly reached and can be only met by either *discarding* newly arriving Interests, or by *overwriting pending Interest state*. Both countermeasures delay content delivery, as can be seen from Figure 6(b). In detail, the time to content delivery of NDN stretches over various PIT combinations up to the final PIT timeout of 10 *s*. It is noteworthy that PIT overflow in these experiments appears for available content that is ready for delivery via valid routes. NDN protocol extensions such as NACKs would neither help nor should be triggered, since Interest retransmissions act as counter measures to packet loss or timeouts due to wireless link degradations. Consequently, the quantitative impairment of packet delivery tightly depends on the scenario and can lead to significant data loss in the constrained IoT, as well.

These experiments shed again light on the trade-off between memory and network performance in the NDN stateful forwarding regime as has been first identified in [60] and recently discussed in the IoT context [51].

4.6 Multi-Hop Topologies

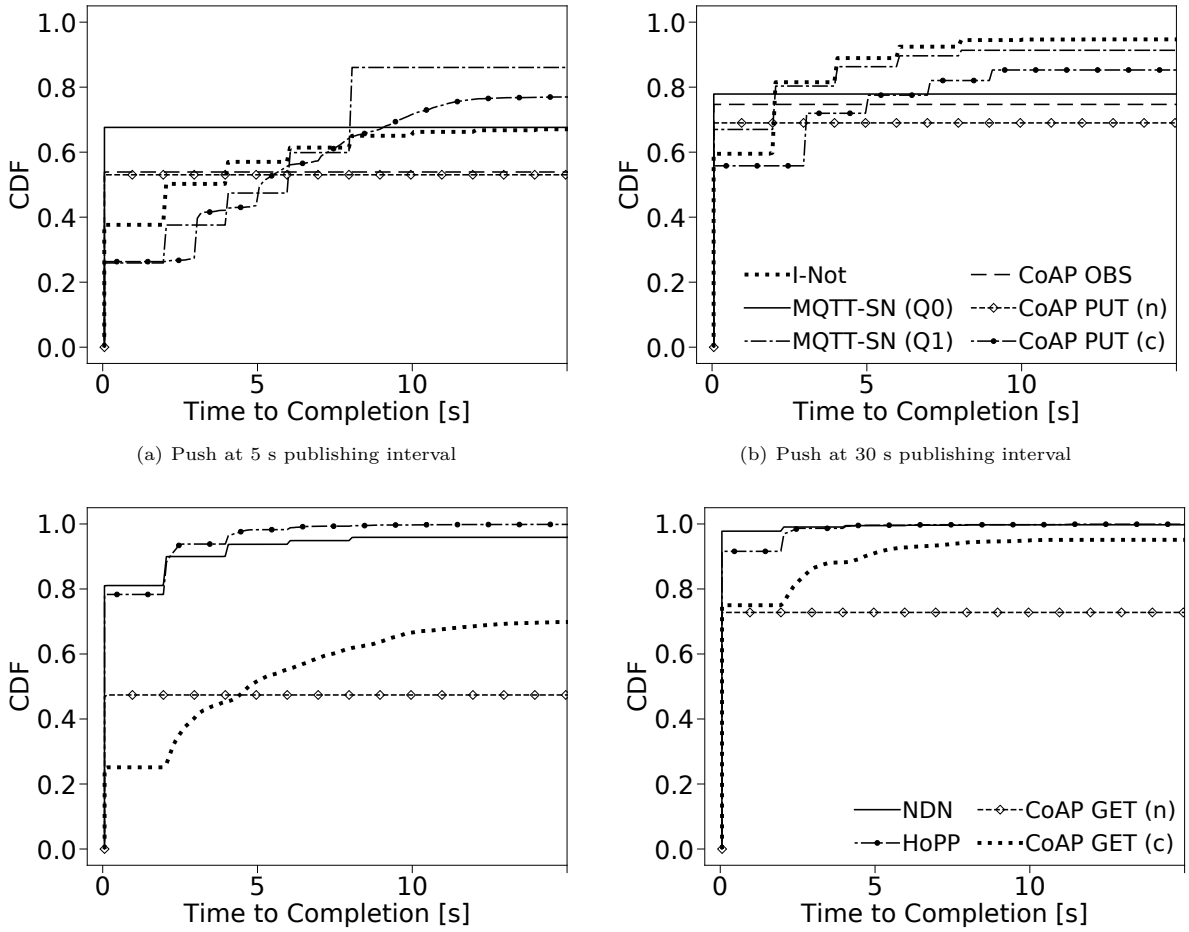
We now consider the more delicate use case of mixed communication in multi-hop topologies: 50 nodes exchange content that is published every 5 or 30 seconds in an uncoordinated manner. Repeated experiments were performed on the Grenoble testbed with tree topologies of routing depths varying from four to six hops.

First, we examine the temporal distributions from content publishing to arrival in analogy to the single-hop cases. Figure 7 combines the results for push and pull protocols, as well as both publishing rates. The overall results reveal a much slower and less reliable protocol behavior than could be expected from the single-hop values in Figure 5. Graphs reflect the common experience in low power multi-hop environments that interferences and individual error probabilities accumulate in a destructive manner.

Push and pull protocols now operate on similar time scales in the absence of considerable disturbances, while events of strong interference and packet corruption on the air lead to large retransmission delays and loss. Protocol retransmissions with an interval of 2 seconds are clearly reflected by the staircase patterns in the respective CDF. Most notably, the 'reliable' variants of CoAP PUT (c) and GET (c) fail to always transfer the content, but remain unsuccessful in a range between 5 % (at 30 s publishing) and 30 % (at 5 s). Even though confirmable CoAP operates more reliably than the unreliable versions OBS and PUT/GET (n), the failure rates indicate a quite unsatisfactory protocol behavior. A similarly unsuccessful performance must be observed for the NDN push variant Interest Notification. In contrast, the reliable MQTT (Q1) successfully transfers its data in 90–95 % of all cases, thereby heavily relying on retransmissions as we will see in the course of the further analysis.

The performance of NDN shows decent results both in promptness and reliability, even though 5 % of data chunks remain lost in the fast publishing scenario (5 s). The only protocol that delivers reasonably fast at full reliability is the NDN variant HoPP. Below we will see that this happens with the least retransmissions and in evenly balanced flows. In a way, this result is not surprising as HoPP is optimized for IoT demands and the only protocol that balances data transmissions per hop. It is the common experience in the low power wireless that link qualities vary quickly and largely.

Second, we evaluate the effective data goodput and flow analysis of the different protocols during content publishing experiments. In Figures 8 and 9, we summarize the results for the variants of NDN, MQTT, and CoAP respectively. We display the different experimental results of the data goodput



(a) Push at 5 s publishing interval

(b) Push at 30 s publishing interval

(c) Pull at 5 s publishing interval

(d) Pull at 30 s publishing interval

Figure 7: Time to content arrival in multi-hop topologies of 50 nodes.

in box plots and compare to the theoretical optimum (lines). Time series of data goodput are further revealing the flow behavior as displayed in the lower row of these figures.

Clearly, HoPP admits the most evenly balanced flows and shows nearly optimal goodput values, closely followed by NDN. All other flow performances fluctuate with some tendency of instability when approaching its full transmission speed. Some IP-based flows in MQTT and CoAP drop to lower delivery rates which is dominantly caused by slow repeated end-to-end retransmission. Multi-hop retransmissions in this error-prone regime tend to cause additional interferences and accumulate transmission errors. As a consequence, protocols operate at reduced efficiency – in some cases protocol performance drops down to 50 % (*e.g.,* CoAP GET (n) and CoAP OBS in Fig. 9). Interest Notification which is not capable of content caching, does not outperform the IP protocols. The overall results show that the absence of flow control as in UDP/IP–based protocols and in the I-Not variant of NDN make protocols fragile. Hop-wise retransmission management as applicable in NDN and HoPP re-balances flows and explicitly demonstrates its benefits for the IoT instead.

Our next evaluation focuses on the link utilization. We measure all individual paths that each unique data packet traveled on its destination from source to sink and contrast the results with the corresponding shortest possible path. Results are visualized as scatterplots in Figure 10. Each dot represents one or several events, the dot size is drawn proportionally to event multiplicities. Solid lines indicate the shortest paths, while events left of the line represent failures (traversal shorter than the shortest path). Right of the solid diagonal retransmissions are counted.

The ideal protocol performance is situated on the diagonal line with all data traversing each link only once on the shortest path. This ideal behavior is most closely approximated by the NDN core and the NDN HoPP protocols. A largely contrasting performance can be seen from the reliable IP protocols MQTT (Q1) and CoAP PUT (c) which admit

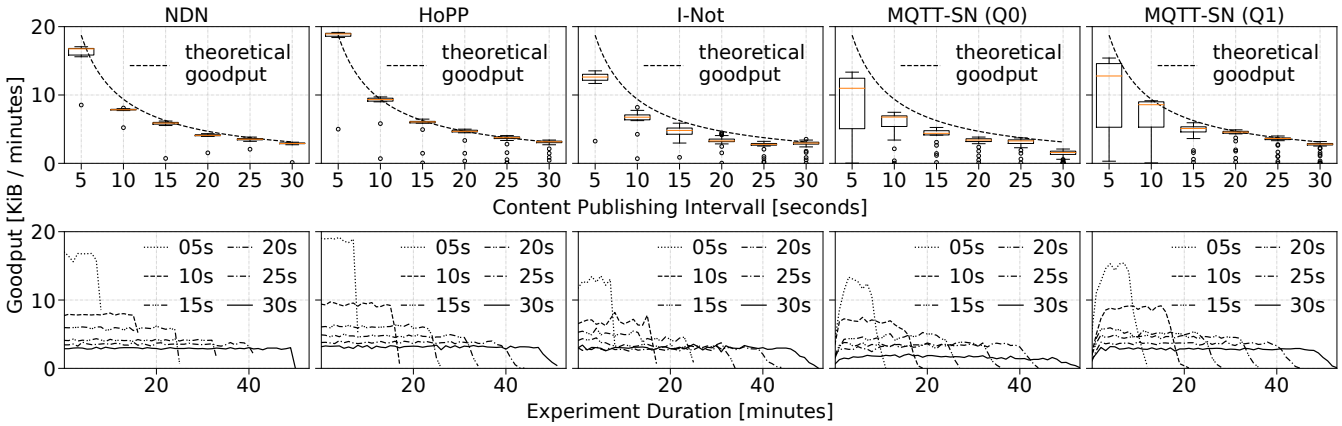


Figure 8: Goodput summary and evolution for the NDN and MQTT protocols at different publishing intervals.

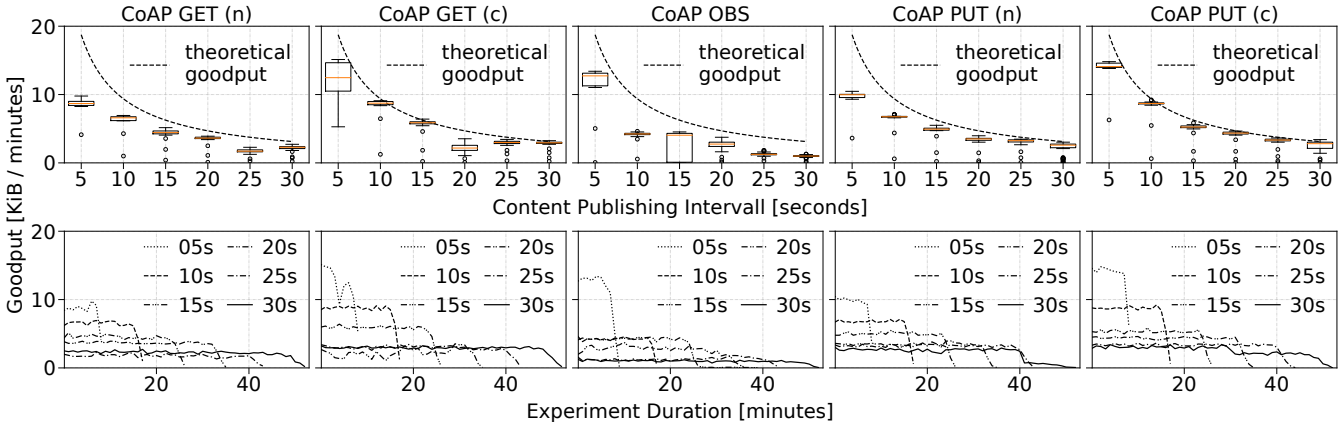


Figure 9: Goodput summary and evolution for the CoAP protocols at different publishing intervals.

huge numbers of retransmissions. This also holds for the NDN Interest Notification protocol which cuts out the NDN feature strength by inverting its semantic.

Unreliable IP-based protocols show very large loss multiplicities and only a few retransmissions which are initiated by reacting to link-layer failures. This corresponds to the reduced success rate already observed in the previous evaluations. Apparently, all protocols that follow an end-to-end path semantic (including I-Not) are forced to struggle against the unpredictable nature of intermediate links—either by voluminous packet retransmissions or significant packet loss.

In our final experimental comparison between the protocols, we evaluate the individual energy consumption per node as a function of time. Since the energy demand of a protocol is largely dominated by its radio transfer of packets, we focus our measurements on 'bytes in the air', i.e., the IEEE 802.15.4 transmission and reception of packets on each individual node. Power consumption levels for transmitting, receiving and idling are obtained from the Atmel AT86RF231 data sheet and we calculate the actual energy from measuring the radio operation time in the respective device state.

Time series of nodal energies are plotted in Figure 11 for each protocol during the course of the experiment. Immediately we observe the tree topology pattern in all graphs: The root node prominently consumes a multiple of leaf node energies, and intermediate forwarders differentiate according to tree ranks in between. It is noteworthy that the routing topology did not rearrange during the measurement period. A varying use of routing trees could gradually balance the uneven energy needs.

Aside from topological effects, distinct protocol signatures become visible. While all energy curves fluctuate due to temporal variations and local retransmissions, some protocols show significant amplitudes from local disorder and repair. Reliable MQTT (Q1) exhibits a peak of recovery after an initial period of loss with depleted energy level on some branch, and a high number of pronounced peaks otherwise.

HoPP experiences a handover in energy load at about eight minutes. This follows its ability of dynamically switching to a more reliable uplink path without rebuilding the topology. HoPP and NDN admit rather steady and smooth energy gradients, since they mainly rely on local repairs (or caching).

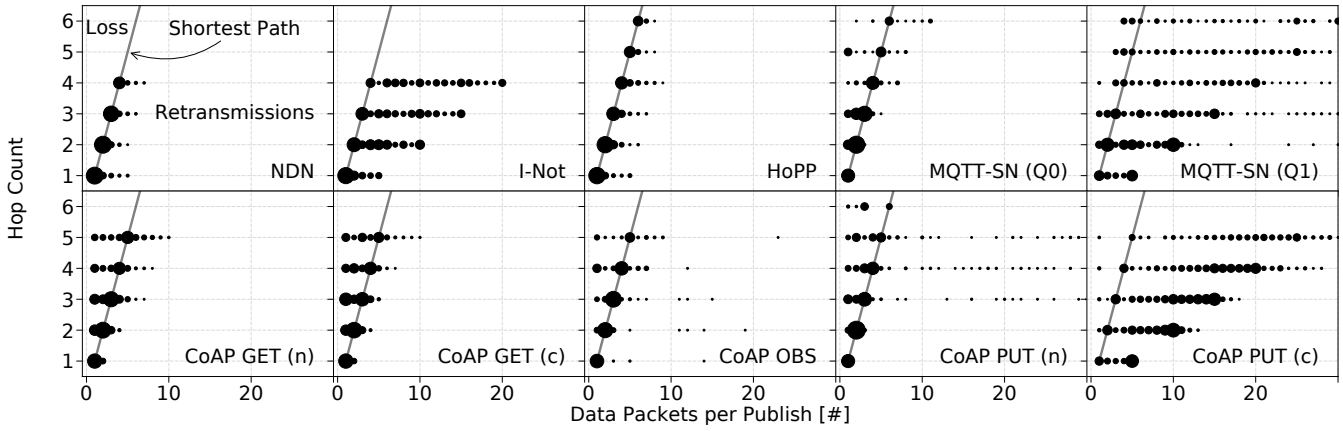


Figure 10: Link traversal vers. shortest path for a 15 s publishing interval. The scatterplots reveal the link stress with dot sizes proportional to event multiplicity.

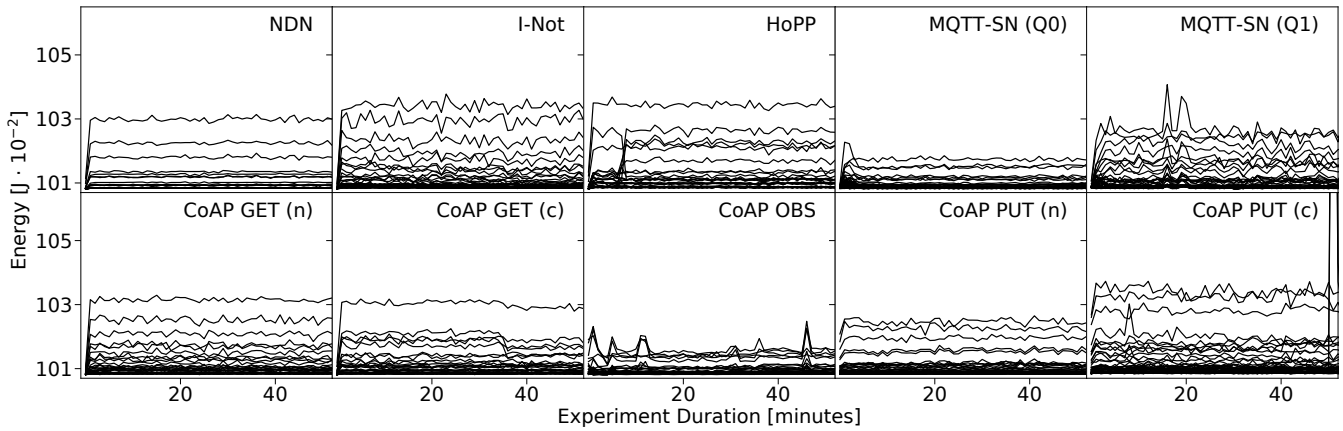


Figure 11: Energy consumption over time for each node in the topology using a 15 s publishing interval.

In contrast, I-Not as a protocol without in-network caching support requires more hop-wise retransmissions and must be considered energy-wise expensive.

Unreliable protocols such as MQTT (Q0) and CoAP (n) repeatedly show valleys in energy curves, since packets lost early on the path relieve the burden of forwarding to the remainders. Delivery failures in CoAP GET (c) as already known from Figure 10 lead to some drops in energy, as well. MQTT (Q0), CoAP OBS and CoAP PUT (n) consume the least energy, which is not surprising for these lean protocols without loss recovery.

Viewing link-stress (Fig. 10) and energy flow (Fig. 11) conjointly, a rather clarifying view on the operational conditions of the protocols emerges. Some protocols remain lean and undemanding while delivering only a restricted service (e.g., CoAP OBS and PUT (n)), others are steady, predictable and run at full service (e.g., NDN and HoPP), and some protocols really struggle in this IoT-typic environment (e.g., MQTT (Q1), CoAP PUT (c), and I-Not).

5 RELATED WORK

5.1 ICN and IoT

The benefits of ICN/NDN in the IoT have been analyzed mainly from three angles. (*i*) design aspects [6, 10, 41, 44, 52], (*ii*) architecture work [21, 49], and (*iii*) use cases [5, 14, 23, 46]. To support experimental evaluation, several implementations have become available, including CCN-lite [58] on RIOT [8, 9] and on Contiki [3], and NDN on RIOT [50]. The objective of this paper is not to present an additional ICN implementation for the IoT but to reuse common stacks. With this we contribute to more reliability of existing software as extensive usage helps to find bugs.

The evaluation of NDN protocol properties in the wild includes the exploitation of NDN communication patterns to improve wireless channel management [26, 27] as well as data delivery on the network layer [10]. Comparison to common IoT network stacks at the transport layer (in particular UDP) is not available. In this paper, we close the gap towards the

application layer and analyze common application protocols (*i.e.,* MQTT and CoAP) compared to intrinsic network layer characteristic provided by NDN.

5.2 Interoperation and Adoption of CoAP and MQTT in ICN

Implementing CoAP on top of ICN has been proposed to enable full features of CoAP [19, 56], such as support of group communication and delay-tolerant communication [30]. These concepts have been showcased in building management systems [20]. In contrast to the integration of CoAP into ICN, an MQTT-to-CCN gateway was proposed to allow for interoperation between CCN IoT devices and the current Internet [3]. A dedicated rendezvous point to discover resources and to bridge between IP-based MQTT subscribers and NDN sensors was introduced in [32]. Note that our work differs from those research as we assess the performance of CoAP, MQTT, and NDN in their original deployment scenarios, instead of focusing on interoperability use cases. This helps to identify intrinsic protocol characteristics.

5.3 Performance evaluation of CoAP and MQTT

The performances of CoAP and MQTT have been studied from several perspectives over the last years [17, 29]. Very early work analyzed the interoperability of specific CoAP implementations [38, 59] without performance evaluation. Later, CoAP implementations have been assessed in comparison to HTTP [39] or on different hardware architectures [35]. MQTT was evaluated in [18]. Thangavel *et al.* [57] proposed a common middleware to abstract from CoAP and MQTT. Based on this middleware, CoAP and MQTT were evaluated in a single-hop wired setup. In emulation, MQTT and CoAP have been studied in the context of medical application scenario [16]. A holistic analysis of MQTT and CoAP in a consistent experimental setting including low-end IoT devices is missing. In particular, no detailed comparison to NDN is provided.

6 CONCLUSIONS AND OUTLOOK

This paper presented extensive experimental analyses to answer the question which of the common protocols MQTT, CoAP, or NDN is best suited for transferring IoT data from constrained devices. We found that for simple, single-hop topologies the protocol families examined in this paper behave similar, but lean push protocols such as MQTT-SN and CoAP Observe operate fastest, at lowest energy consumption, and most network-friendly.

In challenged multi-hop scenarios, though, the results quickly turn tides into a differentiated view between protocols that operate in host-to-host semantic and those acting per link traversal. NDN and NDN-HoPP can both enfold their hop-wise transfer features in balancing flows that reliably deliver data without the need for remarkable retransmission rates. This is in significant contrast compared to common UDP-based IoT application layer protocols that do not benefit from underlying flow control.

While NDN is susceptible of overflowing PIT states in unscheduled publishing scenarios, NDN-HoPP handles such notification events without any performance flaw. In contrast, all IP-based protocols and also the NDN Interest Notification quickly struggle in challenging regimes, either by losing or by repeating packets at large scale.

Our overall findings clearly indicate that lean and simple protocols such as MQTT and CoAP Observe can enfold its efficiencies in relaxed environments with low error rates. Challenged networks, though, will quickly degrade their performance to a minimum. In disruptive environments, protocol performance improves with operations confined to the local link: Hop-by-hop transfer with intermediate caching notably increases reliability and reduces corrective actions, which jointly grants efficient robustness. Dependable systems in challenged regimes should take advantage of corresponding solutions.

With these results, we hope to contribute insights to the community and to strengthen deployment in the constrained IoT. Our future work will concentrate on progressing, deploying, and measuring distributed data systems in the IoT domain that will grant operational insights from real-world deployment and at the same time foster an open, innovative, and robust Internet of Things.

A Note on Reproducibility

We explicitly support reproducible research [1, 47]. Our experiments have been conducted in an open testbed. The source code of our implementations (including scripts to setup the experiments, RIOT measurement apps etc.) will be available on Github at https://github.com/5G-I3/ACM-ICN-2018.

Acknowledgments

We thank our shepherd Dave Oran and the anonymous ICN reviewers for their careful feedback and constructive guidance, which significantly helped to improve the paper. This work was supported in parts by the German Federal Ministry of Research and Education within the project *I3: Information-centric Networking for the Industrial Internet.*

REFERENCES

[1] ACM. Jan., 2017. Result and Artifact Review and Badging. http://acm.org/publications/policies/artifact-review-badging.
[2] Bengt Ahlgren, Christian Dannewitz, Claudio Imbrenda, Dirk Kutscher, and Börje Ohlman. 2012. A Survey of Information-Centric Networking. *IEEE Communications Magazine* 50, 7 (July 2012), 26–36.
[3] Bengt Ahlgren, Anders Lindgren, and Yanqiu Wu. 2016. 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, 221–222.
[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] M. Amadeo, C. Campolo, A. Iera, and A. Molinaro. 2015. 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, 648–653.
[6] Onur Ascigil, Sergi Reñé, George Xylomenos, Ioannis Psaras, and George Pavlou. 2017. A Keyword-based ICN-IoT Platform. In

Proc. of 4th ACM Conference on Information-Centric Networking (ICN). ACM, New York, NY, USA, 22–28.

[7] Atmel. 2009. *Low Power 2.4 GHz Transceiver for ZigBee, IEEE 802.15.4, 6LoWPAN, RF4CE, SP100, WirelessHART, and ISM Applications*. Atmel Corporation. http://www.atmel.com/images/doc8111.pdf

[8] 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. *The IEEE Internet of Things Journal* (2018). http://dx.doi.org/10.1109/JIOT.2018.2815038

[9] Emmanuel Baccelli, Oliver Hahm, Mesut Günes, Matthias Wählisch, and Thomas C. Schmidt. 2013. RIOT OS: Towards an OS for the Internet of Things. In *Proc. of the 32nd IEEE INFOCOM. Poster*. IEEE Press, Piscataway, NJ, USA, 79–80.

[10] 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)*. ACM, New York, 77–86. http://dx.doi.org/10.1145/2660129.2660144

[11] Andrew Banks and Rahul Gupta (Eds.). 2014. *MQTT Version 3.1.1*. OASIS Standard. OASIS. http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html

[12] C. Bormann. 2014. *6LoWPAN-GHC: Generic Header Compression for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)*. RFC 7400. IETF.

[13] C. Bormann, M. Ersue, and A. Keranen. 2014. *Terminology for Constrained-Node Networks*. RFC 7228. IETF.

[14] Jeff Burke, Paolo Gasti, Naveen Nathan, and Gene Tsudik. 2013. 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, 394–398.

[15] Jiachen Chen, Mayutan Arumaithurai, Lei Jiao, Xiaoming Fu, and Kadangode Ramakrishnan. 2011. 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, 99–110.

[16] Yuang Chen and Thomas Kunz. 2016. Performance evaluation of IoT protocols under a constrained wireless access network. In *International Conference on Selected Topics in Mobile & Wireless Networking (MoWNeT)*. IEEE, Piscataway, NJ, USA, 1–7.

[17] Jasenka Dizdarevic, Francisco Carpio, Admela Jukan, and Xavi Masip-Bruin. 2018. *Survey of Communication Protocols for Internet-of-Things and Related Challenges of Fog and Cloud Computing Integration*. Technical Report 1804.01747. ArXiv e-prints.

[18] Asma Elmangoush, Ronald Steinke, Thomas Magedanz, Andreea Ancuta Corici, Alex Bourreau, and Adel Al-Hezmi. 2015. Application-derived communication protocol selection in M2M platforms for smart cities. In *Proc. of 18th International Conference on Intelligence in Next Generation Networks (ICIN)*. IEEE, Piscataway, NJ, USA, 76–82.

[19] Nikos Fotiou, Hasan Islam, Dmitrij Lagutin, Teemu Hakala, and George C. Polyzos. 2016. CoAP over ICN. In *Proc. of IFIP NTMS*. IEEE, Piscataway, NJ, USA, 1–4.

[20] Nikos Fotiou, George Xylomenos, George C. Polyzos, Hasan Islam, Dmitrij Lagutin, Teemu Hakala, and Eero Hakala. 2017. ICN Enabling CoAP Extensions for IP Based IoT Devices. In *Proc. of ACM ICN*. ACM, New York, NY, USA, 218–219.

[21] J. J. Garcia-Luna-Aceves. 2017. 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 (IoTDI '17)*. ACM, New York, NY, USA, 27–36.

[22] Mark Gritter and David R. Cheriton. 2001. An Architecture for Content Routing Support in the Internet. In *Proc. USITS'01*. USENIX Association, Berkeley, CA, USA, 4–4.

[23] Cenk Gündogan, Peter Kietzmann, Thomas C. Schmidt, Martine Lenders, Hauke Petersen, Matthias Wählisch, Michael Frey, and Felix Shzu-Juraschek. 2017. 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, 214–215.

[24] 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)*. IEEE Press, Piscataway, NJ, USA. Accepted for publication.

[25] Cenk Gündogan, Thomas C. Schmidt, Matthias Wählisch, Christopher Scherb, Claudio Marxer, and Christian Tschudin. 2018. *ICN Adaptation to LowPAN Networks (ICN LoWPAN)*. IRTF Internet Draft – work in progress 02. IRTF. https://tools.ietf.org/html/draft-gundogan-icnrg-ccnlowpan

[26] Oliver Hahm, Cédric Adjih, Emmanuel Baccelli, Thomas C. Schmidt, and Matthias Wählisch. 2016. 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, 195—196. http://dx.doi.org/10.1145/2984356.2985226 Best Poster Award.

[27] Oliver Hahm, Emmanuel Baccelli, Thomas C. Schmidt, Matthias Wählisch, Cedric Adjih, and Laurent Massoulié. 2017. Low-power Internet of Things with NDN and Cooperative Caching. In *Proc. of 4th ACM Conference on Information-Centric Networking (ICN)*. ACM, New York, NY, USA, 98–108.

[28] K. Hartke. 2015. *Observing Resources in the Constrained Application Protocol (CoAP)*. RFC 7641. IETF.

[29] Markel Iglesias-Urkia, Adrián Orive, and Aitor Urbieta. 2017. Analysis of CoAP Implementations for Industrial Internet of Things: A Survey. *Procedia Computer Science* 109 (2017), 188–195.

[30] Hasan Islam, Dmitrij Lagutin, and Nikos Fotiou. 2017. Observing IoT Resources over ICN. In *Proc. of IFIP Networking Workshop on Information-Centric Fog Computing*. IEEE, Piscataway, NJ, USA, 1–8.

[31] Van Jacobson, Diana K. Smetters, James D. Thornton, and Michael F. Plass. 2009. Networking Named Content. In *5th Int. Conf. on emerging Networking Experiments and Technologies (ACM CoNEXT'09)*. ACM, New York, NY, USA, 1–12.

[32] José Quevedo and Rui Ferreira and Carlos Guimarães and Rui L. Aguiar and Daniel Corujo. 2017. Internet of Things discovery in interoperable Information Centric and IP networks. *Internet Technology Letters* 1 (2017), 1–6. Issue 1.

[33] 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.

[34] Teemu Koponen, Mohit Chawla, Byung-Gon Chun, Andrey Ermolinskiy, Kye Hyun Kim, Scott Shenker, and Ion Stoica. 2007. A Data-Oriented (and beyond) Network Architecture. *SIGCOMM Computer Communications Review* 37, 4 (2007), 181–192.

[35] Carel P. Kruger and Gerhard P. Hancke. 2014. Benchmarking Internet of things devices. In *Proc. of 12th IEEE International Conf on Industrial Informatics (INDIN)*. IEEE, Piscataway, NJ, USA, 611–616.

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

[37] Martine Lenders, Peter Kietzmann, Oliver Hahm, Hauke Petersen, Cenk Gündogan, Emmanuel Baccelli, Kaspar Schleiser, Thomas C. Schmidt, and Matthias Wählisch. 2018. *Connecting the World of Embedded Mobiles: The RIOT Approach to Ubiquitous Networking for the Internet of Things*. Technical Report arXiv:1801.02833. Open Archive: arXiv.org. https://arxiv.org/abs/1801.02833

[38] Christian Lerche, Klaus Hartke, and Matthias Kovatsch. 2012. Industry adoption of the Internet of Things: A constrained application protocol survey. In *Proc. 17th IEEE International Conf on Emerging Technologies & Factory Automation (ETFA)*. IEEE, Piscataway, NJ, USA, 1–6.

[39] Alessandro Ludovici, Pol Moreno, and Anna Calveras. 2013. TinyCoAP: A Novel Constrained Application Protocol (CoAP) Implementation for Embedding RESTful Web Services in Wireless Sensor Networks Based on TinyOS. *J. Sensor and Actuator Networks* 2, 2 (2013), 288–315.

[40] A. Markkanen and D. Shey. 2015. *Edge Analytics in IoT*. Technical Report. ABI Research.

[41] 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.
[42] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler. 2007. *Transmission of IPv6 Packets over IEEE 802.15.4 Networks*. RFC 4944. IETF.
[43] S. Y. Oh, D. Lau, and M. Gerla. 2010. Content Centric Networking in tactical and emergency MANETs. In *2010 IFIP Wireless Days*. IEEE, Piscataway, NJ, USA, 1–5.
[44] 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.
[45] E. Rescorla and N. Modadugu. 2012. *Datagram Transport Layer Security Version 1.2*. RFC 6347. IETF.
[46] Divya Saxena, Vaskar Raychoudhury, and Nalluri SriMahathi. 2015. SmartHealth-NDNoT: Named Data Network of Things for Healthcare Services. In *Proc. of Workshop on Pervasive Wireless Healthcare (MobileHealth)*. ACM, New York, NY, USA, 45–50.
[47] Quirin Scheitle, Matthias Wählisch, Oliver Gasser, Thomas C. Schmidt, and Georg Carle. 2017. Towards an Ecosystem for Reproducible Research in Computer Networking. In *Proc. of ACM SIGCOMM Reproducibility Workshop*. ACM, New York, NY, USA, 5–8.
[48] Thomas C. Schmidt, Sebastian Wölke, Nora Berg, and Matthias Wählisch. 2016. 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, 458–466.
[49] E. M. Schooler, D. Zage, J. Sedayao, H. Moustafa, A. Brown, and M. Ambrosin. 2017. 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, 1717–1728.
[50] 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.
[51] 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.
[52] Wentao Shang, Yingdi Yu, Teng Liang, Beichuan Zhang, , and Lixia Zhang. 2015. *NDN-ACE: Access Control for Constrained Environments over Named Data Networking*. Technical Report NDN-0036. NDN.
[53] Z. Shelby, K. Hartke, and C. Bormann. 2014. *The Constrained Application Protocol (CoAP)*. RFC 7252. IETF.
[54] Junxiao Shi and Beichuan Zhang. 2012. *NDNLP: A Link Protocol for NDN*. NDN, Technical Report NDN-0006. NDN Team.
[55] Andy Stanford-Clark and Hong Linh Truong. 2013. *MQTT For Sensor Networks (MQTT-SN) Version 1.2*. Protocol Specification. IBM. http://mqtt.org/new/wp-content/uploads/2009/06/MQTT-SN_spec_v1.2.pdf
[56] Sridhar Srinivasa Subramanian, Joseph Pasquale, and George C. Polyzos. 2017. CoAP for Content-Centric Networks. In *Proc. of IEEE CCNC*. IEEE, Piscataway, NJ, USA, 467–472.
[57] Dinesh Thangavel, Xiaoping Ma, Alvin Valera, Hwee-Xian Tan, and Colin Keng-Yan Tan. 2014. Performance evaluation of MQTT and CoAP via a common middleware. In *Proc. of ISSNIP*. IEEE, Piscataway, NJ, USA, 1–6.
[58] Christian Tschudin, Christopher Scherb, et al. 2018. CCN Lite: Lightweight implementation of the Content Centric Networking protocol. http://ccn-lite.net
[59] Berta Carballido Villaverde, Dirk Pesch, Rodolfo de Paz Alberola, Szymon Fedor, and Menouer Boubekeur. 2012. Constrained Application Protocol for Low Power Embedded Networks: A Survey. In *Proc. of 6th International Conf on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*. IEEE Computer Society, Washington, DC, USA, 702–707.
[60] Matthias Wählisch, Thomas C. Schmidt, and Markus Vahlenkamp. 2012. Bulk of Interest: Performance Measurement of Content-Centric Routing. In *Proc. of ACM SIGCOMM, Poster Session*. ACM, New York, 99–100. http://conferences.sigcomm.org/sigcomm/2012/paper/sigcomm/p99.pdf
[61] Matthias Wählisch, Thomas C. Schmidt, and Markus Vahlenkamp. 2013. Backscatter from the Data Plane – Threats to Stability and Security in Information-Centric Network Infrastructure. *Computer Networks* 57, 16 (Nov. 2013), 3192–3206. http://dx.doi.org/10.1016/j.comnet.2013.07.009
[62] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, JP. Vasseur, and R. Alexander. 2012. *RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks*. RFC 6550. IETF.
[63] 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.
[64] Minsheng Zhang, Vince Lehman, and Lan Wang. 2017. Scalable name-based data synchronization for named data networking. In *IEEE INFOCOM'17 (INFOCOM '17)*. IEEE Computer Society, Los Alamitos, CA, USA, 1–9.