Toward a RESTful Information-Centric Web of Things: A Deeper Look at Data Orientation in CoAP

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

Christian Amsüss
christian@amsuess.com

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

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

ABSTRACT

The information-centric networking (ICN) paradigm offers replication of autonomously verifiable content throughout a network, in which content is bound to names instead of hosts. This has proven beneficial in particular for the constrained IoT. Several approaches, the most prominent of which being Named Data Networking, propose access to named content directly on the network layer. Independently, the IETF CoAP protocol group started to develop mechanisms that support autonomous content processing and in-network storage.

In this paper, we explore the emerging CoAP protocol building blocks and how they contribute to an information-centric network architecture for a data-oriented RESTful Web of Things. We discuss design options and measure characteristic performances of different network configurations, which deploy CoAP proxies and OSCORE content object security, and compare with NDN. Our findings indicate an almost continuous design space ranging from plain CoAP at the one end to NDN on the other. On both ends—ICN and CoAP—we identify protocol features and aspects whose mutual transfer potentially improves design and operation of the other.

CCS CONCEPTS

• **Networks** → **Network protocol design**; **Web protocol security**; **Network reliability**; *Network experimentation.*

KEYWORDS

Internet of Things, ICN, CoAP Proxy, OSCORE, content object security, protocol evaluation

ACM Reference Format:
Cenk Gündoğan, Christian Amsüss, Thomas C. Schmidt, and Matthias Wählisch. 2020. Toward a RESTful Information-Centric Web of Things: A Deeper Look at Data Orientation in CoAP. In *ACM Conference on Information-Centric Networking (ICN '20), September 29–October 1, 2020, Virtual Event, Canada.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3405656.3418718

ICN '20, September 29–October 1, 2020, Virtual Event, Canada

ACM ISBN 978-1-4503-8040-9/20/09…$15.00
https://doi.org/10.1145/3405656.3418718

1 INTRODUCTION

More than a decade ago Information-Centric Networking (ICN) [5, 61] introduced the idea to turn named content objects into first class citizens of the Internet ecosystem. This new paradigm gave rise to (i) a decoupling of content from hosts and thus the ability of ubiquitous content caching [4] without a clumsy, closed CDN (Content Delivery Network) infrastructure, and (ii) serverless routing on names without the DNS infrastructure [21]; (iii) Named Data Networking (NDN) [28, 62] additionally abandoned network endpoint addresses in favor of a stateful forwarding fabric. These properties enable an asynchronous, hop-by-hop content fetching, which prevents forwarding of unwanted data. The latter significantly reduces the attack surface of (Distributed) Denial-of-Service (DDoS).

All three constituents make ICN appealing to the (constrained) Internet of Things (IoT) as infrastructural burdens and common DDoS threats, which have established in the current Internet, stand in the way of a lean and efficient inter-networking for embedded devices. Early experimental work [12, 37] could indeed show that NDN can successfully operate on very constrained nodes with noticeable resource savings compared to IP. In addition, short-term in-network caching proved valuable for increasing reliability in low power lossy networks with nodes frequently at sleep as common at the IoT edge [23, 26].

Since that time, the Internet of Things is gaining momentum and its deployment is driven by industrial needs [25]. These needs are served by the protocol interfaces available from cloud providers—predominantly MQTT [13] (such as Amazon AWS)—or by the IETF IoT protocol suite centered around the Constrained Application Protocol (CoAP) [52]. The CoAP protocol group (CoRE) has recently developed a rich set of additional features, which open various deployment options—content object security and in-network caching are among them.

In this paper, we explore the emerging building blocks of the CoAP protocol suite to answer the question: *Can we build a restful Web of Things that adheres to ICN first hand principles and performance?* We carefully explore in quality and quantity each CoAP protocol element in comparison to NDN and discuss how they can contribute to an information-centric IoT. We define scenarios that range from plain CoAP deployment over several extended settings that include content object security, proxying, and caching, and evaluate their performance in detail. Our findings indicate that the design space between end-to-end CoAP and the hop-wise content replication of NDN is almost continuously populated when

combining selected CoAP protocol extensions with appropriate configurations.

Our insights reveal that the available CoAP building blocks including CoAP proxies with caches and content object security with OSCORE are nearly complete for building a RESTful Information-Centric Web of Things. Particular protocol functions in both, the CoAP and the NDN world, could be identified as potential protocol enhancement when transferred to the other world. In following this approach, we enliven the hope to take advantage of the various insights and techniques that emerged from ICN research and lead them into a promising, realistic deployment trail for the fast emerging IoT.

The remainder of this paper is structured as follows. We discursively summarize the problem space with related work in Section 2. Section 3 introduces the new feature set of the CoAP world and qualitatively compares various protocol aspects with ICN. In Section 4, we develop the five deployment scenarios that span the current design space. These scenarios are quantitatively evaluated in Section 5 from the perspective of showing both protocol performance as well as design nuances in a meaningful multi-hop setting of a real-world IoT testbed. We discuss our findings and insights in Section 6, including lessons learned for future design work, and conclude in Section 7.

2 THE PROBLEM OF BUILDING AN INFORMATION-CENTRIC IOT AND RELATED WORK

The Internet of Things (IoT) increasingly connects embedded controllers built into intelligent machines and at the same time drives a huge deployment of sensor devices, which collect and report measurements from the wild. This massive machine-to-machine communication exchanges syntactically and semantically well-structured data for further aggregation and processing in some cloud. This data-centric nature at the Internet edge called for rethinking the current IoT architecture [49], and emphasized consideration of information-centric principles in the future IoT development.

Coping with Constraints. The mass constituents of the IoT will be tiny, cheap *things* that communicate via low power and often lossy channels. The IETF has designed a suite of protocols that adapt to this constrained environment. The IPv6 adaptation layer 6LoWPAN [35] enables a deployment on constrained links (e.g., IEEE 802.15.4), which RPL routing arranges in a multi-hop topology [60]. The Constrained Application Protocol (CoAP) [52] offers a lightweight alternative to HTTP while running over UDP, or DTLS [42] 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 linking the *things*.

ICN networks have been early identified as a lean and efficient network alternative for a future IoT [7, 12, 32]. Popular operating systems for low end IoT devices such as Contiki [18] and RIOT [11] have been providing NDN network stacks [6, 51] for years. ICN-LoWPAN [24], an adaptation layer for NDN and CCNx for constrained wireless links, has been designed and outperforms 6LoWPAN. Hence from a resource perspective, the information-centric concepts and software solutions have well met the challenges posed by the low end IoT edge.

Adapting Communication. Many IoT access networks are wireless, slow, and error-prone. In this context, the original end-to-end design of the Internet [43], which pushes service functions such as reliability up to the transport, turns into a challenge: Several retransmissions via multiple hops quickly exhaust network resources and interfere with subsequent communication requests.

Name-based routing, hop-by-hop forwarding, and in-network caching have shown to support robustness of application scenarios in regimes of low reliability and reduced infrastructure (e.g., without DNS). In comprehensive experiments, network caches established as efficient retransmission buffers, which significantly decreased network load and improved the overall network performance [23]. Several cache optimization strategies for an information-centric IoT [39, 40] could improve the overall network performance and resilience even further.

Securing Content Objects. Adding security credentials to content objects instead of transmission channels is a new approach to secure communication on the Internet. Information Centric Networking first introduced content object security on the network layer for the sake of ubiquitous caching. Recently, the IETF Core working group released OSCORE, which extends the IoT ecosystem to content object security.

OSCORE [50] is a protocol extension to CoAP and addresses the issue of security terminating at gateways. Instead of securing sessions between endpoints, OSCORE protects entire CoAP messages and provides integrity, authenticity, and confidentiality on an object level. The original CoAP message is thereby encapsulated as an authenticated and encrypted COSE [44] object by an outer CoAP option. OSCORE utilizes the request-response semantics of its underlying CoAP layer and an elaborate nonce construction to obtain compact response messages. Recently it was shown that OSCORE message protection clearly outperforms DTLS session security in the constrained IoT and approximates the NDN performance in several dimensions [22].

Shaping a Mainstream Technology. Many forces drive the current development of the IoT and lead to a rather fragmented protocol landscape. Historic domain-specific (local) protocols, traditional industry standards, and the present IETF suite all persist in specific deployments. The traditional request-response content access is the popular approach for the current IoT [17]. It is foreseeable that soon a standard solution will be desired to ensure interoperability between the steadily emerging new applications and deployments. With this in mind, Fotiou et al. [19] developed a CoAP emulation that runs over ICN. Keeping ICN as the underlying network preserves beneficial concepts and technologies that have been developed over the past dozen of years.

The alternative approach is to transfer the insights, design elements, and features established in ICN research to the current IETF standards and transform the protocol composition and its deployment into an ICN variant. We will show in the following that the CoAP protocol suite is almost ready to host an information-centric web of things while complying to the well established Internet standards.

3 COAP VERSUS ICN: A FEATURE SET COMPARISON

3.1 Security

Request-Response Binding. A key aspect of ICN is its ability to address content objects instead of traditional network endpoints. In the most prominent ICN expressions, CCNx and NDN, names bind irrevocably to content. This immutability allows for a range of positive effects, including a long liveliness with regard to caching purposes, a resource-friendly content provenance validation using digests [38], and a desensitization of applications to delayed and replayed messages. Since content requests are considered idempotent, a transactional request-response binding is not required.

CoAP follows the RESTful model and is architecturally akin to HTTP. Requests contain URIs that resolve to service endpoints, which can serve static or dynamic content. Request methods add further semantics to requests and allow for state transitions in the application. Non-cryptographic tokens in the CoAP header match responses to corresponding requests. With security mechanisms layered below CoAP (*e.g.,* the widely deployed datagram transport layer security DTLS [42]), applications need to actively manage their tokens to fend off attacks. Otherwise, the inability to provide a verifiable request-response mapping can be fatal, especially in cases where resources publish mutable content [33]. The OSCORE security layer establishes verifiable message binding, and the upcoming Request-Tag option [8] extends it to fragmented request representations.

Object-level Provenance and Encryption. The integral caching component of ICN systems enables content retrieval from potentially untrusted peers. On that account, most ICN solutions implement data integrity, provenance, and origin authentication on the protocol level [30]. Access control, authorization, and privacy on the other hand are challenged by this pluralistic networking approach and are left to upper layers or the application. In CCNx and NDN systems, security measures are generally applied to returning response messages. Both architectures also allow for the inclusion of digital signatures in request messages.

CoAP by itself does not include any security measures, but was designed like HTTPS to rely on transport layer security by (D)TLS. As a protocol extension, OSCORE protects entire CoAP messages and provides integrity, authenticity, and confidentiality on an object level. The original CoAP message is thereby encapsulated as an authenticated and encrypted compressed COSE [44] object.

Résumé. *ICN authenticates content independent of its consumers, whereas CoAP OSCORE binds security to individual access requests by authenticating and encrypting CoAP messages.*

3.2 In-Network Caching

Cache Model. The immutability of content objects and a name-based routing as applied by CCNx and NDN allow for a seamless integration of on-path content caching in the network. While a ubiquitous caching with adequate cache replacement strategies reduces access times of popular content, it offers one additional benefit that is strikingly valuable especially in lossy environments: caches serve as retransmission buffers in order to boost the content delivery reliability. Retransmissions generally happen on the scale of seconds, *i.e.,* allocated cache space is short-lived and quickly released.

CoAP proxy endpoints [52, Section 5.7] can store messages on two conceptually separate layers[1], in message deduplication and in an application layer cache. Each networked device along a path can operate as a proxy, which will generate a cache distribution similar to ICN.

Messages secured by OSCORE are strictly bound to a single request. Hence, they can only be meaningfully retained in CoAP proxies for message retransmissions. Proxies are not allowed to see details of content as required to find suitable cache entries from previous transmissions. Clients—even the same client served by an older response—lack the context to decrypt it. Efforts to adapt OSCORE group communication to produce cacheable requests are underway, but have not yet produced testable results.

Content Freshness Model. CoAP uses a freshness model that is comparable to the content freshness handling of CCNx and NDN. A CoAP Max-Age option in responses provides a lifetime hint for caching endpoints, after which this response is marked as stale.

Content Validation Model. CoAP applies an efficient validation model to revalidate stale responses using the ETag [52, Section 5.10.6] option in request messages. Instead of transmitting the full response, a validating origin server merely responds with a small message to indicate whether a cached response is considered to be valid again. In contrast, CCNx and NDN have no notion of invalid cache entries, since named content is immutable and can only expire, but not change.

Résumé. *ICN binds names to immutable content for long-term, in-network caching, whereas CoAP proxies cache on a message level, including optimized signaling for validation.*

3.3 Request Handling and Forwarding

Message Synchronization. The ICN design decision to address content independent of its location complicates the temporal decoupling of request and response messages. In name-based routing architectures, the requesting and requested endpoints are unknown. Responses travel along a reverse path that is temporarily constructed from the request. Long intervals between request and response require equally long-lived soft-states on each hop in the network. NFN [54] and RICE [29] are two protocol extensions that support a handling of long-running requests, but long-lived Interests place a burden onto the network.

Plain CoAP deployed between endpoints requires state only at these endpoints. Conversely to CCNx and NDN, the reception of requests is acknowledged by the content producer. Such open request at the consumer can easily be long-lived and allows the producer to respond proactively as soon as content is available.

Reliable Transport. Both protocol families support retransmissions following message timeouts initiated by the requester. For the ICN protocols, retransmissions are not bound to endpoints but happen from hop to hop. If previous requests have populated the on-path caches, retransmissions benefit from cache hits, which pull the content closer to the requester.

[1]The unified design of CoAP as a single protocol spanning both cache layers allows caching at one layer to be foregone in many cases.

Following the host-centric paradigm, CoAP uses end-to-end retransmissions, but can deploy caching proxy nodes to enhance reliability of the transport. On-path caches rebuild a hop-wise content replication and thereby benefits of ICN.

ICN and CoAP support similar features to report on error cases. CoAP encodes error codes into response messages analogous to HTTP. CCNx specifies an Interest Return message and NDN delegates the error reporting to NDNLP [53].

Next-hop Selection. ICN designs typically use content names to perform next-hop lookups in a Forwarding Information Base (FIB). In the common end-to-end CoAP deployment, requests are forwarded based on a destination IP address matched against a FIB.

When CoAP is used with proxies, forwarding decisions are performed on the application level. RESTful Web protocols have established mechanisms to include forward proxies in a network autoconfiguration using WPAD (Web Proxy Auto-Discovery Protocol) [20] and to decide the next-hop based on the host name of the resource using the PAC (Proxy Auto-Config) feature, which is implemented in all common web browsers. No such mechanism has yet been described for the IoT, but the application of analogous techniques seems plausible. Such a mechanism could in particular be used to learn the next-hop from the underlying discovery protocol as a forward proxy, if it was discovered that the capability is available there.

The forwarding decision is usually based on the authority component of the request URI. That typically, but not necessarily contains a resolvable host name. Nodes that cannot resolve an authority component (*e.g.,* because they do not implement DNS) often rely on a default proxy that handles name resolution for them.

DoS Protection. A central design aspect of NDN was to prevent the submission of unwanted content, which has the beneficial effect of making traditional Denial of Service (DoS) impossible [28]. For this, one important building block is the absence of endpoint addresses, which makes it harder to target packets to a specific node. In a dense deployment of CoAP proxies (*i.e.,* a proxy on each forwarding node) very similar techniques can apply. For the next-hop proxy, nodes only need to resolve its link-local address from the FIB, which in turn will be elided by the 6LoWPAN header compression – hence leaving the packet without network address.

Unfortunately, it was soon discovered that stateful Interest forwarding in NDN can lead to a different kind of DoS attack [58, 59], which was later coined 'Interest Flooding' [3]. CoAP proxies are susceptible of similar state inflation attacks.

Résumé. *Both ICN and CoAP with dense proxy deployment can perform a request routing on names (URIs) and a stateful content forwarding. A cache-assisted reliable transport option is available for both families.*

3.4 Multi-Source & Multi-Destination

Multicast. Support for multicast communication is an inherent property of most ICN implementations. The absence of endpoints in the addressing scheme allow for multi-source and multi-destination classes of applications with virtually no added overhead. In popular ICN systems, multi-source communication is designed by aggregating requests at nodes on intersecting request paths. Returning responses fan out to the corresponding requesters. Multi-destination requests are supported due to on-path caches and multiple target entries for the same name prefix in the forwarding information base. Responses that return from multiple destinations are dropped at path intersections as soon as existing request states are consumed by the first response.

CoAP supports group communication using IP multicast [15] as the underlying data transmission [16, 41]. In contrast to the more nuanced multicast integration of ICN designs, CoAP disallows confirmable multicast messages. Retransmissions in case of message timeouts are thus delegated to the application. Current research [55] looks into leveraging proxy nodes to support fan-outs of unicast messages at the proxy and takes up on the open question of how to handle multiple returning responses. One approach is to leave the deduplication of multiple returning messages to the application, while another approach is to aggregate multiple responses at the proxy to return a collective message to the requester. This technique would not only be applicable to routable multicast addresses, but also to proxies that have multiple forward routes for a given resource and authority URI component, allowing setups analogous to ICN architectures with multiple destinations for a prefix.

Mobility. Multicast mobility is an asymmetric problem [47]. While the movement of receivers is often easy to compensate by local network reconfigurations, the impact of mobile sources on the routing is complex and requires assisting measures or services. In a network setup with proxy nodes, multicast proxy services have proven useful in orchestrating network reconfiguration [46, 48], as they adapt locally with only link-local signaling on the control plane. It is expected that these techniques can be transferred to CoAP proxies in a straight-forward manner. Mobility in ICN [57, 63] sees the analogous problem space. It is easily supported for consumers and difficult to implement for content providers.

Protected Group Messages. Object security as commonly implemented in ICN-based systems integrates with the intrinsic multicast support and allows for a seamless group communication with secured messages. The responsibility for a proper key management is entrusted to the deployment.

CoAP is commonly deployed with DTLS in order to provide secure communication channels between endpoints. The end-to-end nature of DTLS complicates a group communication by design. OSCORE brings object security, but the strong binding between the request and response excludes a multicast operation. Ongoing research [56] extends the OSCORE model to tolerate source authentication for CoAP group requests and the corresponding responses.

Résumé. *ICN and (unprotected) CoAP support multicast communication, whereas multi-party communication for proxy-assisted CoAP is still in its design phase.*

4 DEPLOYMENT SCENARIOS

We deploy NDN and different compositions of CoAP protocols as schematized in Figure 1. Starting with plain CoAP `GET` requests, we gradually add more and more protocol features of ICN-nature to approach the NDN setup. Protocol operations and configurations are detailed in the following.

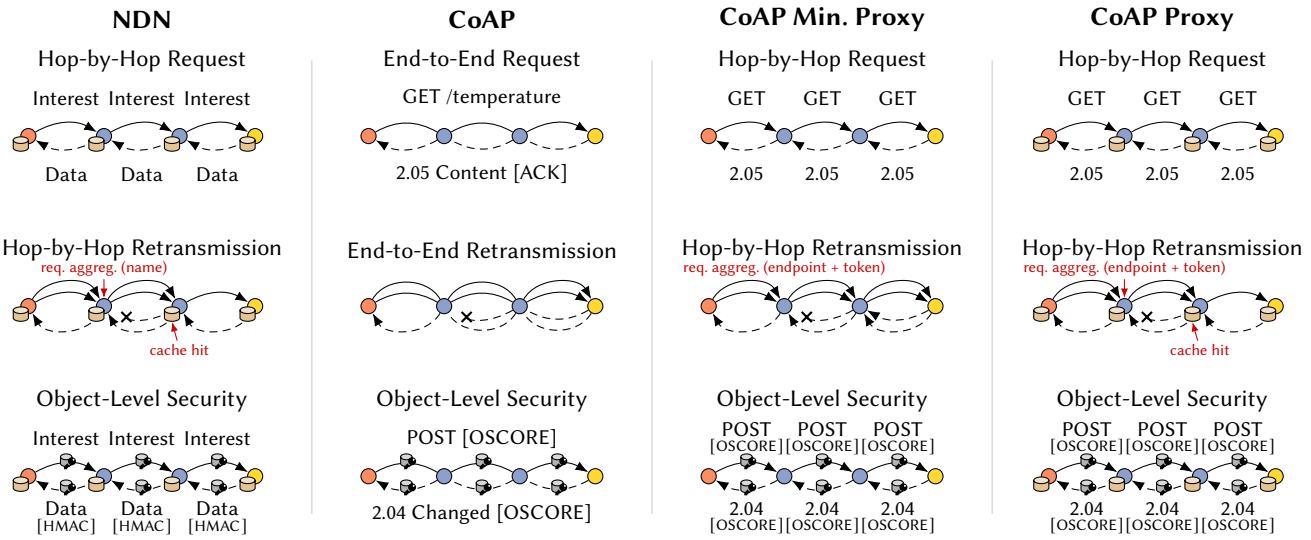

Figure 1: Deployment scenarios and protocol configurations used in our comparative evaluations.

4.1 Standard NDN

Hop-by-Hop Request. Common NDN deployment uses a name-based routing, hop-wise requests, and on-path caches.

Hop-by-Hop Retransmission. Each hop on a request path arms a retransmission timer for Interests. If content is not timely returned, then the initial Interest is repeated. NDN integrates message deduplication and request aggregation features in order to suppress the transmission of Interests for request paths that are already set up.

Object-Level Security. Security on an object level is inherent to NDN. While the outer response packet can be signed using different cryptographic algorithms, an HMAC signature seems most appropriate for the IoT. Encrypting the content within responses is left to the application.

4.2 Routed CoAP

End-to-End Request. CoAP supports different request methods, from which `GET` compares best to Interest requests of NDN. Unlike in NDN, request state exists only at the endpoints.

End-to-End Retransmission. `GET` requests can be issued unreliably (`NON`) and with corrective actions enabled (`CON`). In `GET CON`, each request requires an acknowledgment, which is piggy-backed in the response message. On absence, a retransmission of the initial request message is triggered.

Object-Level Security. OSCORE provides a secure communication between two endpoints. `GET` requests are nested into COSE objects and are cryptographically secured. These objects are then included in CoAP `POST` messages as OSCORE objects. The returning response is treated similarly by nesting the message into a COSE object and delivering the OSCORE object in a `2.04 Changed` response.

4.3 CoAP with Minimal Proxy

Hop-by-Hop Request. CoAP proxy nodes operate at the application level and handle conversions between CoAP and other protocols. A proxy runs as a `reverse`, or a `forward` proxy and is commonly situated at the network edge. Requests that traverse a proxy intermediately terminate and lose their end-to-end semantics between endpoints. Responses follow the same request path through the proxy node in reverse—a property which is well-known from ICN approaches, such as NDN.

In this scenario, we install forward proxies on all forwarder nodes. The minimal version in this scenario is included for illustrative purposes, and lacks message deduplication and storage as is regularly required with CoAP.

CoAP clients include `Proxy-URI` options in request messages to provide forwarding hints to the proxies. This option contains the URI string that encodes the URI scheme, the authority component that identifies the CoAP server, and the service path. Each proxy manages forwarding state and passes requests either to subsequent proxies, or to the origin server. In case the request arrives at the final proxy node, the message is translated for normal CoAP operation, *i.e.,* the Proxy-URI string is split into its URI components and a common `GET` request is transmitted to the origin server.

Hop-by-Hop Retransmission. The CoAP specification does not fully outline the proxy operation for request retransmissions, but we envision the following two scenarios: First, a proxy acknowledges the reception of the request using an empty acknowledgment message and thus pauses any further retransmissions of the previous hop. Second, a proxy identifies incoming retransmissions based on the token and endpoint information. It then aggregates duplicate requests to the outstanding request state. Concurrently, the proxy handles its own retransmissions. For our deployment setup, we

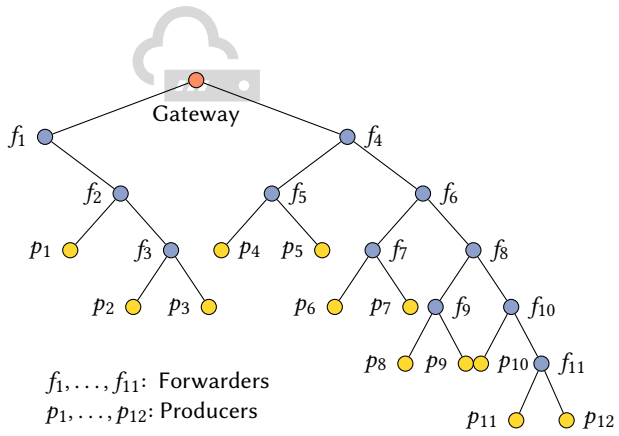

Figure 2: Topological arrangement of gateway, forwarders, and producer nodes for each deployment.

consider the latter approach as it approximates the NDN operation quite well.

Object-Level Security. Messages secured with OSCORE require no additional interaction on proxy nodes. As with various other options, the OSCORE option is copied to the reissued request and thus forwarded until it reaches the designated endpoint. The requested service path name resides within the encrypted security envelope of OSCORE and is not accessible from the outer CoAP message. This does not only protect the authenticity of request-response exchanges from attempts of tampering and forgery, but also retains privacy by hiding the requested path from eaves-droppers. To maintain these security properties, Proxy-URI strings outside the security envelope only contain the URI scheme and authority sections, but not the service path. In secured OSCORE deployments, CoAP proxies thus make forwarding decisions based on less information than in unsecured deployments.

4.4 CoAP with Proxy

Hop-by-Hop Request. The addition of retransmission caches to each forward proxy on a path advances the protocol transformation: This deployment shows huge similarities with NDN in terms of hop-wise message passing and hop-wise caching. A CoAP message deduplication module aggregates requests based on the message correlation parameters. From those, it determines whether a response is already being processed, and does not forward it.

Hop-by-Hop Retransmission. In our setup, no separate responses [52, Section 5.2.2] are used. Thus, the responsibility for ensuring that the response arrives stays with the client. The response content is cached in the proxy at least as long as request retransmissions by the client are expected.

Object-Level Security. OSCORE messages are encapsulated in CoAP `POST` requests and `2.04 Changed` responses. Those are stored in the retransmission cache.

5 EVALUATION IN THE TESTBED

In this section, we quantitatively assess the five deployment scenarios outlined in Section 4 using real protocol implementations and experiments in a testbed.

5.1 Experiment Setup

Use Case and Topology. Our experiments follow a typical IoT application: A consumer node is situated at the network edge (the gateway) and retrieves sensory data (*e.g.,* temperature readings) from content producers. A set of forwarder nodes provides connectivity between the consumer and producers. The gateway, forwarder, and producer nodes statically arrange on system startup in a Destination Oriented Directed Acyclic Graph (DODAG) as illustrated in Figure 2 for all protocol deployments. DODAGs are optimized for the predominant converge cast scenario, *i.e.,* they yield shortest paths from sensors to cloud services, but show suboptimal paths for sensor to sensor traffic. Our setup has a total of 12 producers and 11 forwarder nodes. This minimal constellation can already show signs of link and memory exhaustion on network stress. In this topology, the caches closer to leaves experience less load, while caches near the gateway show cache replacements much more frequently.

Deployment Parameters. In our experiments, the gateway periodically issues requests via the IoT stub network to its edge sensors. Each sensor device is requested 500 times at an interval of 1.25s ± 0.1s and returns a 2-byte temperature value. That time was chosen such as to create a situation of pronounced network load for all scenarios. All experiments are aligned with respect to retransmission and timeout configurations. On message timeout, nodes wait two seconds before initiating a retransmission of the initial request; retransmissions are limited to five. In this work, we do not add explicit interferences from external cross-traffic. Still, each individual transmission experiences background traffic from ongoing requests and retransmissions that are self-induced by the experiment. Requests are jittered, though, to mix the event space and to allow for a better state exploration.

Software & Hardware Platform. All devices run RIOT [10] version 2020.04. NDN deployments are based on CCN-lite and CoAP experiments use the default GNRC network stack of RIOT including libOSCORE[2]. The CoAP forward proxy is an additional software module and was extended for caching.

We conduct our evaluations on the FIT IoT-LAB [2] testbed. The testbed hardware consists of class 2 devices [14] featuring an ARM Cortex-M3 MCU with 64 kB of RAM and 512 kB of ROM. To operate on the IEEE 802.15.4 radio, each device is equipped with an Atmel AT86RF231 [9] transceiver.

5.2 Message Overhead

We first dissect the details of request and response messages of the examined protocols in Figure 3. We fix the response payload to a 2-byte temperature value.

The maximum physical layer packet size of IEEE 802.15.4 is 127 bytes. In our interface configuration, the total MAC header

[2] https://gitlab.com/oscore/liboscore

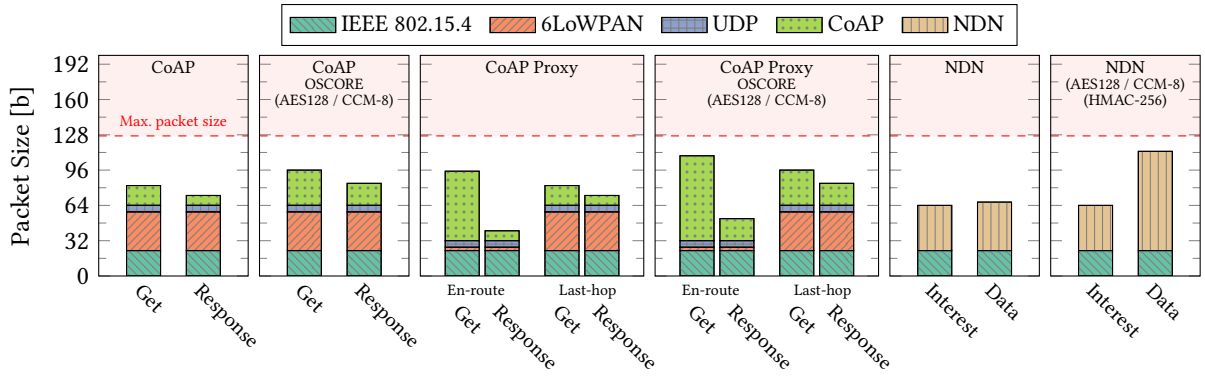

Figure 3: Packet structures and sizes of data-plane packets for each protocol configuration.

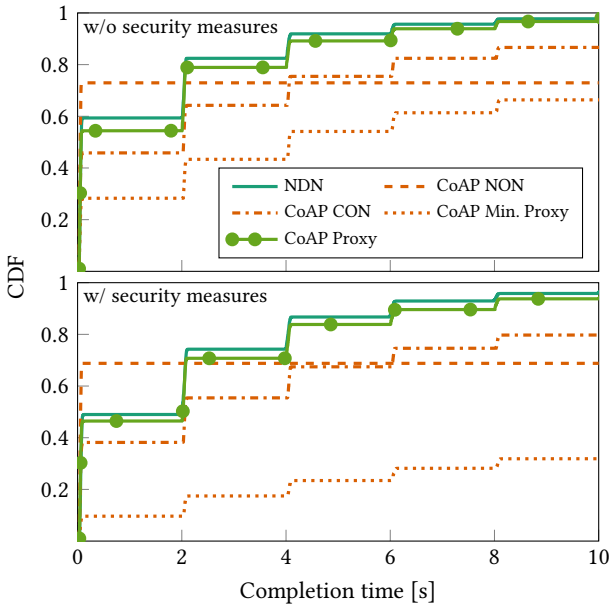

Figure 4: Time to content arrival—distributions for different protocol deployments and security settings.

overhead adds up to 23 bytes, leaving 104 bytes payload size for upper layer protocols.

In all CoAP deployments, the 6LoWPAN overhead accounts for 35 bytes, which carry the dispatch types and two global IPv6 addresses. A single exception are packets forwarded between CoAP proxies: they can use link-local IPv6 addresses as discussed in Section 3.3, which 6LoWPAN can elide by header compression. In this case, the 6LoWPAN overhead reduces to the remaining three dispatch bytes. The compressed UDP header requires an additional 6 bytes.

Request messages in the standard CoAP deployment require 18 bytes on the application layer, which includes the resource URI string `/temperature` and CoAP related protocol information, such as the 2-byte message ID and the 2-byte token. In contrast, response messages display the much smaller packet size of 9 bytes. This is a result of omitting resource URIs in the response and use the 2-byte token to match returning responses to corresponding requests.

Content object security with OSCORE deployment inflates request messages by 14 bytes and response messages by 11 bytes due to security encoding overhead and a message authentication code. An OSCORE protocol optimization allows the same nonce values for cryptographic operations on requests and responses. With this, the nonce value is completely omitted from response messages, as they are obtained from the request state on the requesting node.

The forward proxy deployment of CoAP uses the `Proxy-Uri` option string in requests to designate an endpoint. In contrast to the plain CoAP, sizes of the CoAP protocol increase by 45 bytes for CoAP requests in the unsecured and secured cases. The last forwarder hop prior to the producer node transforms the Proxy-URI string into appropriate CoAP options. Since at the same time messages on the last hop use a global IPv6 destination address, 6LoWPAN needs to include the additional 32 bytes for the addresses again. Response messages do not include any forwarding hints and compare to response sizes of the regular CoAP deployment.

NDN keeps requests smallest with a protocol overhead of 41 bytes. This includes the name. Due to its design that mirrors request names back in Data messages, responses tend to exceed the packet sizes of their requests. The secured variants inflate the message sizes by 46 bytes for Data messages, which is significantly more expensive than OSCORE. Interest messages are not affected by security measures and do not include a security overhead.

5.3 Time to Content Arrival

We measure the times to complete a content request, *i.e.*, the time from requesting content to its arrival at the gateway. Note that this metric summarizes not only the speed of protocol data transmission, but also the distribution of loss events and the effectiveness of corrective protocol actions. Figure 4 displays the corresponding distributions for our compared protocol deployments with and without content object security in place.

We first observe that all protocol families are in rough agreement with the configured retransmission intervals. Distributions in the sub-second range represent transmissions that succeed within one round-trip. Retransmissions operate in a two-second interval and

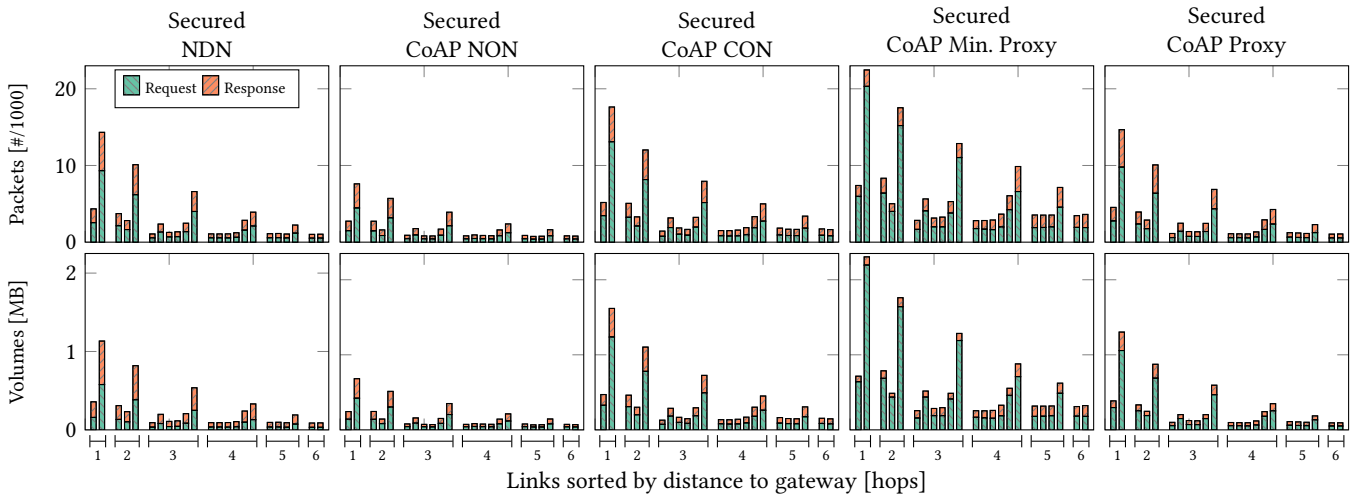

Figure 5: Number of packets and bytes transmitted over the air per downstream and upstream link in the topology for each deployment with security measures enabled.

lead to the stair case pattern observed for all protocols, but CoAP NON. The unreliable CoAP NON protocol is able to successfully complete more requests (≈ 75%) than any other protocol on the first try, which in turn is due to its unreliability: The lack of retransmission control keeps the medium free of retransmissions, hence leaving more capacity to the initial packet transfer. Content security overhead reduces the success of CoAP NON to ≈ 66%.

The reliable protocols CoAP CON and hop-wise CoAP minimal proxy similarly fail in completing the sensor readings within five retransmissions. CoAP minimal proxy operations yield a rather poor temporal distribution with final success rates of 70% (w/o security) and 30% (w/ security). In this setup, the increased packet sizes, but foremost the hop-wise retransmission requests amplify the link stress immensely to a point, where no reliable communication between producers and the gateway is possible. In contrast, CoAP CON shows higher success rates than CoAP minimal proxy due to a lower retransmission control overhead: End-to-end retransmissions sequentially traverse all hops of a path until they reach a destination, or a packet loss occurs. With hop-wise retransmissions, messages originate independently of the previous hop, as long as forwarding state exists from previous attempts.

In contrast, the full CoAP proxy deployment exhibits a success rate of 98 % and performs very similar to NDN. The secured versions show temporal performances that match the distributions on the unsecured cases. Due to the increased message sizes, success rates decrease minimally for all protocols, except for the hop-wise CoAP minimal proxy operation. It is clearly visible that the full CoAP proxy can leverage the potentials of hop-by-hop transfer with intermediate retransmissions served by the caches just as NDN does.

5.4 Link Stress

The topology in Figure 2 generates different levels of link stress for regular communication throughout the network. We measure packet events and total bytes over the air for each protocol and link in the topology using independent sniffer devices. Note that our links are within an overlapping broadcast domain with mutual interferences. Since our experiments use carrier sensing of the radios within a static topology, we argue that our measurements of captured unicast traffic between device pairs serve as a proper estimate on the protocol induced link stress.

Figure 5 displays the results for the secured protocol variants. All links are grouped by their hop distance to the gateway node and we further distinguish between request (link downstream) and response (link upstream) packets. At first, different values in each link group are due to the number of nodes in the sub-tree served by each link.

CoAP NON displays the least amount of packets and even lower data volumes on each link, which is expected due to its lack of retransmission capabilities and smaller packets. All other protocol scenarios show slightly more request packet events than responses. Hop-wise CoAP minimal proxy in particular generates a much large number of request messages than responses. This is due to many request retransmissions triggered by intermediate proxies and corresponds to our observations in the completion time measurements (see Figure 4).

NDN and the full CoAP Proxy show similar results of captured packet events per link and a similar relation between requests and responses. Data volumes, however, differ noticeably: Posing a request is much cheaper in NDN than in CoAP due to the packet structure given in Figure 3. This uneven link utilization is the result of (i) unsecured Interests, which keeps requests small for NDN, and (ii) the additional 32-byte HMAC and the message authentication code for the NDN payload, whereas OSCORE displays a much smaller security footprint.

5.5 Cache Utility

We now confront cache utilization with packet loss on each hop for the secured NDN and full CoAP Proxy. These metrics disclose how efficiently transmission failures can be compensated by a nearby

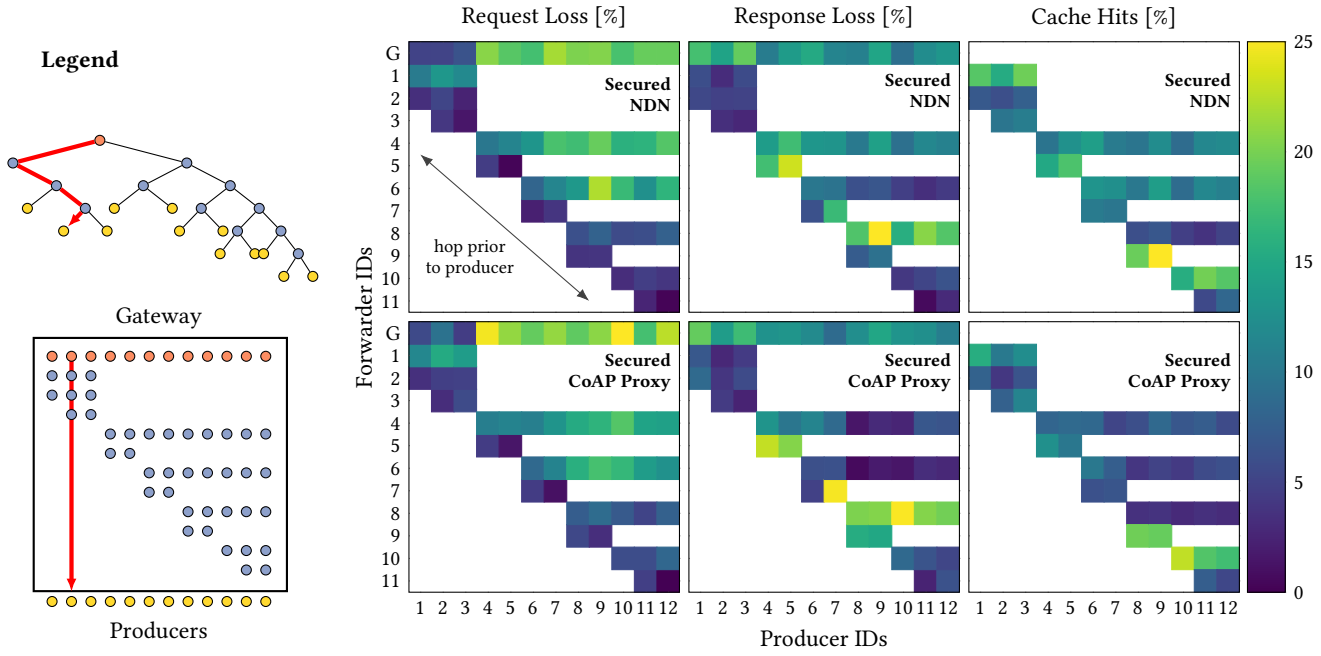

Figure 6: Packet loss and cache hit ratios for secured NDN and CoAP Proxy. Columns represent request paths from top (gateway) to bottom (producer) as illustrated in Figure 2.

cache. Results are displayed in Figure 6. Each cell in the matrix describes the message exchange between a node and its downstream neighbor toward a particular producer. A column from top to bottom represents a valid request path from the gateway to a producer across a varying number of forwarders as illustrated in Figure 2.

We first observe the request losses per link, which cannot be compensated from caches. The overall picture reveals that NDN better succeeds in delivering requests to the next hop, which is expected due to the smaller request message sizes (see Figure 3). CoAP clearly shows to be at a disadvantage with higher efforts in delivering requests, reaching relative loss rates up to 20–25%.

On the message response side, the converse holds: NDN performs slightly worse compared to CoAP, which again can be attributed to the increased message sizes of NDN Data. Looking at the cache hits, we are interested in how efficient caches compensate for data loss. Ideally, a lost response can be served from the next-hop cache on the path, *i.e.,* a bright cell in the loss matrix is shadowed by an equal brightness at the next populated cell of lower Y-coordinate.

Caching services nicely work for NDN: Response losses on the line $Y = 4$ for example are compensated by the caches on line $Y = 6$ (the next populated), and the high loss at coordinate (9,8) is immediately serviced from the next cache (9,9). Cache services are less pronounced for CoAP, since data losses are less pronounced. Also by accident, one lossy link (7,7) directly connects to a producer without intermediate cache. On the overall CoAP shows a fair cache utility, as well.

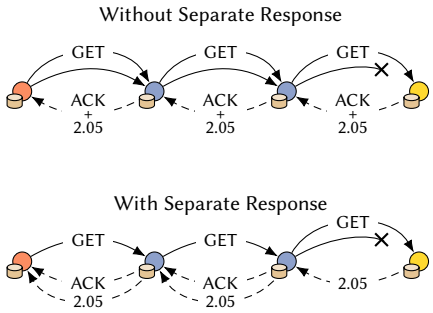

Figure 7: The mechanism of early acknowledgments as separate response to reduce hop-wise request retransmissions.

5.6 Early Acks in Separate Responses

Confirmable CoAP messages are retransmitted until an acknowledgment arrives, or a message timeout occurs. For confirmable requests, CoAP allows to piggyback acknowledgments in returning data responses. This is the preferred mode if data responses are generated immediately. Separate response [52, Section 5.2.2] is a protocol enhancement in CoAP to pause unnecessary request retransmissions of the client in case the response generation takes longer than the configured request message timeout. In this mode, an empty acknowledgment message is promptly sent to the requesting client. Once content is available, a response with the actual content is then returned.

In this evaluation, we want to quantify the control overhead of the secured CoAP proxy deployment and compare it to a deployment variant that uses separate responses as illustrated in Figure 7: each CoAP proxy is configured to immediately acknowledge an

incoming GET request, while the origin server responds with a piggybacked acknowledgment as before.

Figure 8 shows the frequency of outgoing requests—including request retransmissions—that originate from the gateway node and incoming responses received by the gateway over the duration of the experiment. Our first observation is that CoAP without an early acknowledgment mechanism and NDN display similar performances. This is coherent with our results in Figure 4 and Figure 5. Both deployments employ the same hop-wise retransmission strategy and show analogous completion times as well as success rates. It is expected that the request to retransmission ratio is also comparable.

In detail, we observe a number of outgoing requests at a rate of 30–35 packets per second for NDN and CoAP without immediate acknowledgments. Roughly 50% of all requests on both links at the gateway node (see Figure 2) consists of request retransmissions. In contrast, separate responses visibly reduce the overall requests from the gateway to below 20 packets per second. Retransmissions represent only ≈ 23% of all requests. The differences are equally pronounced when observing the request to retransmission ratio across the entire network: For CoAP without early acknowledgments and NDN, ≈ 40% of total requests in the network are retransmissions. With the use of early acknowledgments, this number reduces to 25% for CoAP.

All protocols exhibit very similar performances when inspecting the amount of incoming responses at the gateway node. However, CoAP without early acknowledgment shows a subtle decline in overall success rates. Around 96% of requests have a corresponding response, while NDN and CoAP with early acknowledgment both display a success rate of 98%. The reduced number of control overhead does not only reduce the utilization of network resources, but also lessens link stress and increases success rates.

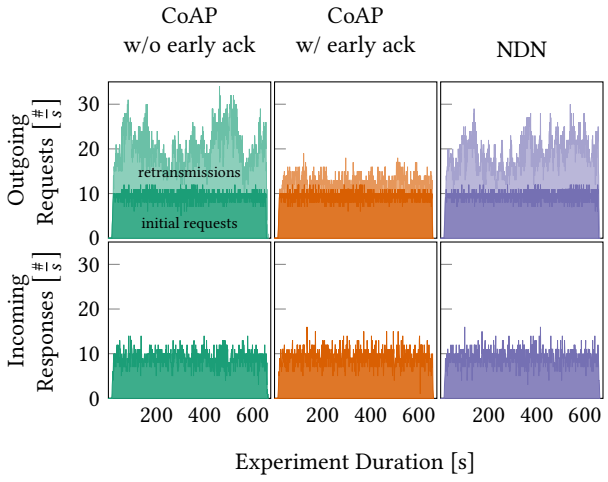

Figure 8: The effect of early acknowledgments in CoAP—numbers of outgoing requests and incoming responses measured at the gateway node.

6 DISCUSSION

Individual protocol components and their interaction can impact performance significantly. We will now discuss how the exchange of protocol building blocks between the worlds impacts corresponding network performance.

CoAP. We have seen during our journey on deploying a CoAP scenario with ICN characteristics that the combination of two building blocks shows substantial performance improvements. Chaining CoAP proxies with caches enables link-scoped corrective actions. This shortens retransmission paths and reduces link traversals in networks with high loss probabilities. The compact handling of link-local addresses, which can be compressed away, is resource efficient and at the same time demonstrates a formal coincidence with the address-less NDN architecture.

The hop-by-hop forwarding between proxy nodes potentially leaks service paths and therefore sensitive data to the application logic. The problem of name confidentiality is also prevalent in ICN architectures and ample approaches have already been proposed in the literature that provide obfuscation mechanisms for routed name prefixes [27, 31]. Due to the high similarities between NDN and CoAP proxy deployments, the obfuscating approaches can easily be adapted to the Proxy-URI string components.

Our study also identified that while a retransmission cache is sufficient to gain ICN-like benefits for a single client, content level caching which serves multiple clients not only requires careful application design, but also poses interesting challenges for OSCORE use cases.

Further message size reduction is possible by using the CoAP split options for expressing the URI, and by using reverse proxying styles. Smaller messages are beneficial because of increased transmission success (see Section 5.3). Moreover, successful *requests* have the additional effect of building a request path which starts populating caches for later use when responses may be lost (as noted in Section 5.5). Our experiments further indicate much higher positive impacts for smaller requests, as they quickly build a request path and profit from hop-wise retransmissions for a response.

As a last point, we want to discuss in-network state for CoAP. The original deployment idea follows the basic packet network concept of stateless forwarding with network state persisting on the endpoints, only. In the information-centric CoAP deployment, all nodes including the forwarders maintain request state. As main memory is constrained in low-power networked devices, the number of open request handles at each node is equally limited. At first sight, the overhead added on each forwarder appears as a disadvantage that may lead to quickly saturating memory resources and denial of service on request paths. Our IoT experiments in NDN deployments, however, show that content caching and request aggregation features are able to limit resource usages immensely by shortening path lengths and reducing completion times of open requests.

ICN. The full CoAP proxy has a similar cache model as CCNx and NDN. Unlike in HTTP, neither protocol family supports cache policy control in request messages. Content producers determine content lifetime values on message creation and requests cannot bypass valid cache entries en-route. CoAP adds an efficient cache

validation model: requests that meet stale cache entries trigger secondary requests to the original server to check on content validity. Returning responses may either include a confirmation of validity or new content. We argue that a cache revalidation model for ICN would optimize bandwidth consumption not only in IoT stub networks and want to pursue its utility in future work.

We see value in adopting separate responses [52, Section 5.2.2] to control the retransmission behavior of previous hops. NDN and CCNx already support an error reporting infrastructure using Interest NACK [34] and Interest Return [36]. These mechanisms could be extended to deploy a similar retransmission control strategy. Adding retransmissions not only to requests, but also to responses is a technique commonly used in CoAP deployments to increase success rates and we suggest an experimental analysis with similar approaches for information-centric deployments.

The CoAP token mechanism seems applicable to reduce response sizes: Requests could carry a short token that maps to a name on each forwarding hop, possibly using the Pending Interest Table (PIT). A similar technique is already employed by the en-route compression functionality of ICNLoWPAN [24]. Instead of mirroring back the full name, responses could include the short token in order to map to the corresponding request on the reverse path. Reducing the response size does not yield the same benefits as reducing the request size, but still reduces a major contributor to the link stress.

7 CONCLUSION & OUTLOOK

In this paper, we presented a conceptual feature comparison between CoAP and archetypal ICN designs. We set out with the motivation to build a RESTful CoAP deployment that inherits information-centric properties and conduct a comprehensive analysis to quantify the effective network performances in the (low-power) Internet of Things.

Our findings indicate that (1) loosening the end-to-end principle, (2) adding retransmission caches, and (3) utilizing object security enables secure, RESTful deployments that achieve comparable network performances as observed with NDN. As a result of compiling a feature compendium for CoAP and NDN, we were also able to identify striking protocol elements that bear potentials to improve protocol operations if transferred from one architecture to the other.

We have shown that the differences in caching and even in naming between original information-centric designs and those originating from an end-to-end mindset are more by convention than by necessity. Assimilating RESTful CoAP deployments towards a named-data networking architecture allows reusing and exploring the impact of many well-studied concepts in new deployment environments. We will focus on these surfacing approaches and emerging properties in our future work.

A Note on Reproducibility

We fully support reproducible research [1, 45] and perform all our experiments using open source software and an open access testbed. Code and documentation is available on Github at https://github.com/inetrg/ACM-ICN-2020-COAP.

Acknowledgments

We want to thank the anonymous reviewers and our shepherd Lixia Zhang for constructive feedback and inspiration on how to improve the paper. 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*, *RAP-store – RIOT App Store*, and the Hamburg *ahoi.digital* initiative with *SANE*. The `libOSCORE` library was made with financial support from Ericsson AB.

REFERENCES

[1] ACM. Jan., 2017. Result and Artifact Review and Badging. http://acm.org/publications/policies/artifact-review-badging.
[2] Cedric Adjih, Emmanuel Baccelli, Eric Fleury, Gaetan Harter, Nathalie Mitton, Thomas Noel, Roger Pissard-Gibollet, Frederic Saint-Marcel, Guillaume Schreiner, Julien Vandaele, and Thomas Watteyne. 2015. FIT IoT-LAB: A large scale open experimental IoT testbed. In *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*. 459–464.
[3] Alexander Afanasyev, Priya Mahadevan, Ilya Moiseenko, Ersin Uzun, and Lixia Zhang. 2013. Interest Flooding Attack and Countermeasures in Named Data Networking. In *Proc. of IFIP Networking*. IEEE Press, Piscataway, NJ, USA.
[4] Bengt Ahlgren, Matteo D'Ambrosio, Marco Marchisio, Ian Marsh, Christian Dannewitz, Börje Ohlman, Kostas Pentikousis, Ove Strandberg, René Rembarz, and Vinicio Vercellone. 2008. Design Considerations for a Network of Information. In *Proc. of Re-Architecting the Internet Workshop (ReARCH)* (Madrid, Spain) *(ReARCH '08)*. ACM, New York, NY, USA, 66:1–66:6.
[5] 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.
[6] 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.
[7] 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.
[8] Christian Amsuess, John Mattsson, and Goeran Selander. 2020. *CoAP: Echo, Request-Tag, and Token Processing*. Internet-Draft – work in progress 10. IETF.
[9] 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
[10] Emmanuel Baccelli, Cenk Gündogan, Oliver Hahm, Peter Kietzmann, Martine Lenders, Hauke Petersen, Kaspar Schleiser, Thomas C. Schmidt, and Matthias Wählisch. 2018. RIOT: an Open Source Operating System for Low-end Embedded Devices in the IoT. *IEEE Internet of Things Journal* 5, 6 (December 2018), 4428–4440. http://dx.doi.org/10.1109/JIOT.2018.2815038
[11] 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.
[12] Emmanuel Baccelli, Christian Mehlis, Oliver Hahm, Thomas C. Schmidt, and Matthias Wählisch. 2014. Information Centric Networking in the IoT: Experiments with NDN in the Wild. In *Proc. of 1st ACM Conf. on Information-Centric Networking (ICN-2014)* (Paris). ACM, New York, 77–86. http://dx.doi.org/10.1145/2660129.2660144
[13] 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
[14] C. Bormann, M. Ersue, and A. Keranen. 2014. *Terminology for Constrained-Node Networks*. RFC 7228. IETF.
[15] Stephen E. Deering and David R. Cheriton. 1990. Multicast Routing in Datagram Internetworks and Extended LANs. *ACM Trans. Comput. Syst.* 8, 2 (1990), 85–110.
[16] Esko Dijk, Chonggang Wang, and Marco Tiloca. 2020. *Group Communication for the Constrained Application Protocol (CoAP)*. Internet-Draft – work in progress 01. IETF.
[17] Jasenka Dizdarevic, Francisco Carpio, Admela Jukan, and Xavi Masip-Bruin. 2019. Survey of Communication Protocols for Internet-of-Things and Related Challenges of Fog and Cloud Computing Integration. *ACM Comput. Surv.* 51, 6 (Jan. 2019), 116–1 – 116–29.
[18] Adam Dunkels, Björn Grönvall, and Thiemo Voigt. 2004. Contiki - A Lightweight and Flexible Operating System for Tiny Networked Sensors.. In *Proc. of IEEE Local Computer Networks (LCN)*. IEEE Computer Society, Los Alamitos, CA, USA, 455–462.
[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] Paul Gauthier, Josh Cohen, Martin Dunsmuir, and Charles Perkins. 1999. *Web Proxy Auto-Discovery Protocol.* Internet-Draft – work in progress 01. IETF.

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

[22] Cenk Gündogan, Christian Amsüss, Thomas C. Schmidt, and Matthias Wählisch. 2020. IoT Content Object Security with OSCORE and NDN: A First Experimental Comparison. In *Proc. of 19th IFIP Networking Conference* (Paris, France). IEEE Press, Piscataway, NJ, USA, 19–27. https://ieeexplore.ieee.org/document/9142731

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

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

[25] Cenk Gündogan, Peter Kietzmann, Thomas C. Schmidt, and Matthias Wählisch. 2020. Information-Centric Networking for the Industrial Internet of Things. In *Wireless Networks and Industrial IoT*, Nurul Huda Mahmood, Nikolaj Marchenko, Mikael Gidlund, and Petar Popovski (Eds.). Springer. https://doi.org/10.1007/978-3-030-51473-0

[26] 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.

[27] Huan he and Bo Chen. 2019. An Elliptic Curve Based Name Privacy Protection Mechanism for Sensory Data Centric Named Data Networking. In *Proc. of 15th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN).* IEEE, Piscataway, NJ, USA, 56–62.

[28] 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)* (Rome). ACM, New York, NY, USA, 1–12.

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

[30] 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.

[31] Tobias Lauinger, Nikolaos Laoutaris, Pablo Rodriguez, Thorsten Strufe, Ernst Biersack, and Engin Kirda. 2012. Privacy Risks in Named Data Networking: What is the Cost of Performance? *SIGCOMM Comput. Commun. Rev.* 42, 5 (September 2012), 54–57.

[32] 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.

[33] John Mattsson, John Fornehed, Goeran Selander, Francesca Palombini, and Christian Amsuess. 2018. *Controlling Actuators with CoAP.* Internet-Draft – work in progress 06. IETF.

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

[35] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler. 2007. *Transmission of IPv6 Packets over IEEE 802.15.4 Networks.* RFC 4944. IETF.

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

[37] 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.

[38] K. Pentikousis, B. Ohlman, E. Davies, S. Spirou, and G. Boggia. 2016. *Information-Centric Networking: Evaluation and Security Considerations.* RFC 7945. IETF.

[39] Jakob Pfender, Alvin Valera, and Winston K.G. Seah. 2019. Easy as ABC: A Lightweight Centrality-Based Caching Strategy for Information-Centric IoT. In *Proceedings of the 6th ACM Conference on Information-Centric Networking* (Macao, China) *(ICN '19).* ACM, New York, NY, USA, 100–111.

[40] Ioannis Psaras, Wei Koong Chai, and George Pavlou. 2012. Probabilistic In-network Caching for Information-centric Networks. In *Proc. of the second ICN workshop on Information-centric networking* (Helsinki, Finland). ACM, New York, NY, USA, 55–60.

[41] A. Rahman and E. Dijk. 2014. *Group Communication for the Constrained Application Protocol (CoAP).* RFC 7390. IETF.

[42] E. Rescorla and N. Modadugu. 2012. *Datagram Transport Layer Security Version 1.2.* RFC 6347. IETF.

[43] Jerome H. Saltzer, David P. Reed, and David D. Clark. 1984. End-to-End Arguments in System Design. *ACM Trans. Comput. Syst.* 2, 4 (Nov 1984), 277–288.

[44] J. Schaad. 2017. *CBOR Object Signing and Encryption (COSE).* RFC 8152. IETF.

[45] 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.

[46] T. Schmidt, S. Gao, H. Zhang, and M. Waehlisch. 2014. *Mobile Multicast Sender Support in Proxy Mobile IPv6 (PMIPv6) Domains.* RFC 7287. IETF.

[47] T. Schmidt, M. Waehlisch, and G. Fairhurst. 2010. *Multicast Mobility in Mobile IP Version 6 (MIPv6): Problem Statement and Brief Survey.* RFC 5757. IETF.

[48] T. Schmidt, M. Waehlisch, and S. Krishnan. 2011. *Base Deployment for Multicast Listener Support in Proxy Mobile IPv6 (PMIPv6) Domains.* RFC 6224. IETF.

[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] G. Selander, J. Mattsson, F. Palombini, and L. Seitz. 2019. *Object Security for Constrained RESTful Environments (OSCORE).* RFC 8613. IETF.

[51] 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.

[52] Z. Shelby, K. Hartke, and C. Bormann. 2014. *The Constrained Application Protocol (CoAP).* RFC 7252. IETF.

[53] Junxiao Shi and Beichuan Zhang. 2012. *NDNLP: A Link Protocol for NDN.* NDN, Technical Report NDN-0006. NDN Team.

[54] Manolis Sifalakis, Basil Kohler, Christopher Scherb, and Christian Tschudin. 2014. An Information Centric Network for Computing the Distribution of Computations. In *Proceedings of the 1st ACM Conference on Information-Centric Networking* (Paris, France) *(ICN '14).* ACM, New York, NY, USA, 137–146.

[55] Marco Tiloca and Esko Dijk. 2020. *Proxy Operations for CoAP Group Communication.* Internet-Draft – work in progress 01. IETF.

[56] Marco Tiloca, Goeran Selander, Francesca Palombini, and Jiye Park. 2020. *Group OSCORE - Secure Group Communication for CoAP.* Internet-Draft – work in progress 09. IETF.

[57] Gareth Tyson, Nishanth Sastry, Ruben Cuevas, Ivica Rimac, and Andreas Mauthe. 2013. A Survey of Mobility in Information-centric Networks. *Commun. ACM* 56, 12 (Dec. 2013), 90–98.

[58] 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* (Helsinki). ACM, New York, 99–100. http://conferences.sigcomm.org/sigcomm/2012/paper/sigcomm/p99.pdf

[59] 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

[60] 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.

[61] George Xylomenos, Christopher N. Ververidis, Vasilios A. Siris, Nikos Fotiou, Christos Tsilopoulos, Xenofon Vasilakos, Konstantinos V. Katsaros, and George C. Polyzos. 2014. A Survey of Information-Centric Networking Research. *IEEE Communications Surveys and Tutorials* 16, 2 (2014), 1024–1049.

[62] 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.

[63] Yu Zhang, Alexander Afanasyev, Jeff Burke, and Lixia Zhang. 2016. A survey of mobility support in Named Data Networking. In *Proc. of IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS).* IEEE, Piscataway, NJ, USA, 83–88.