On the cost of geographic forwarding for information-centric things

Marcel Enguehard, *Student Member, IEEE*, Ralph Droms, and Dario Rossi *Senior Member, IEEE*

***Abstract*—Recent research has identified Information-Centric Networking (ICN) as a good fit for Internet of Things (IoT) deployments. However, most studies have focused on ICN as an application enabler, disregarding the behaviour from a network viewpoint. In this paper, we address this by studying the most important properties of an ICN-IoT deployment and contrast the operational costs between geographic-based forwarding and name-based forwarding schemes. We aim to understand if, and under which IoT deployment characteristics, geographic forwarding constitutes an advantage over name-based schemes, in terms of feasibility (i.e., memory footprint and computational capability of the devices) and performance (which we analyze as the overall energy cost of operating an ICN-IoT network under either forwarding paradigm).**

To achieve this goal, we employ a mixture of (i) modelling, (ii) simulative and (iii) experimental methodologies, which are useful to respectively (i) state the problem in a principled way, (ii) gather information about topological properties that are instrumental to the model and (iii) gather physical properties of the devices to feed the model with realistic data. In a nutshell, our results show that geographic forwarding (i) halves the memory footprint on our reference deployments and (ii) yields significant energy savings, especially for dynamic topologies.

I. Introduction

Recent research [1]–[12] argues that Information-Centric Networking (ICN), a network paradigm centered around the concept of named information available in the network, is a better fit for the Internet of Things (IoT) than the standard IPv6-based network stack. For instance, seminal work [1] shows through experiments that a slightly modified ICN stack outperforms the standard IPv6-based stack (i.e., IEEE 802.15.4, 6LoWPAN and RPL) in terms of energy efficiency and memory requirements – which are primary concerns for IoT. The study of the ICN paradigm has focused both on IoT at large [2], [3] as well as on specific use cases such as vehicular networks [4], industrial automation [5], healthcare [6], smart cities [7], buildings [8] and homes [9].

However, the advantages resulting from the use of ICN architectures in terms of naming, mobility, and security [2] do not come without challenges [3]. One such challenge is efficient packet delivery with minimal control traffic. Geographic forwarding, where packets are forwarded towards a physical location instead of a host, addresses this issue by keeping routing updates local. Yet, geographic forwarding strategies have not been explored in ICN-IoT in detail [4], [10], and require further attention. Security aspects constitute another challenge: IoT data is often private (e.g., position [4] or health sensor readings [6]) and must be protected from malicious attackers. Additionally, the IoT should be protected from malicious nodes performing denial of service (DoS) attacks (e.g., flooding malicious packets to drain the IoT relays batteries [10]). The challenge is thus to ensure that the ICN security model is fit to provide network access control [11], [12] on power-constrained IoT nodes.

In this paper, we focus on the first of these two challenges: how to transmit secured IoT sensor data efficiently over the network using geographic forwarding. We refer the reader to our previous work [10], [13] for more details on the articulation of security and efficient forwarding in the ICN-IoT. Specifically, we are interested in assessing whether these two goals can be jointly achieved in a way that is not only feasible but also, and especially, energy efficient. While geographic forwarding strategies have been thoroughly studied in the literature since the publication of seminal work such as [14], to date there are no implementations available in state of the art IoT stacks [15], and the same holds true for the ICN-IoT context. We are thus interested in understanding if, and under which circumstances, secure geographic forwarding can be a viable alternative to address-based (in IP) or name-based (in ICN) forwarding for the general IoT applications. Rather than focusing on a specific IoT deployment, which would result in conclusions of limited scope, we intentionally study the broad issue of *geographic* as opposed to *name-based* forwarding, where resource consumption grows with the number of neighbours and entries in the ICN forwarding information base (FIB) respectively. At the same time, to make the analysis relevant from a practical viewpoint, we also single out four real deployments [16]–[19] as a reference, which represent diverse scenarios: environmental monitoring, smart cities, buildings, and homes.

To summarize our contributions:

- we propose a simple analytical model to represent the energy consumption of ICN-IoT systems supporting name-based and geographic forwarding schemes;
- we implement a functioning RIOT-based prototype of ICN-IoT, which we plan to make available as open source in the future, and gather accurate data on message encryption and transmission at individual devices, as one source of data to drive the model;
- we simulate message propagation dynamics over large topologies, from which we gather propagation patterns as a second source of data to the model;
- finally, given an energy budget, we use the model to derive the expected number of messages for different degrees of network dynamism under both schemes, giving useful guidelines for ICN-IoT deployments.

The remainder of this paper is organized as follows. We start

TABLE I
REFERENCE IOT DEPLOYMENTS

	Deployment name	Deployment class	No. of Nodes	Node degree	Ref.
A	Place de la Nation	Urban sensor network	97	3.8	[16]
B	Great Duck Island	Environmental sensor network	150	4.6	[17]
C	CASAS	Home automation	30	8	[18]
D	Sensor Andrew	Building automation	1000	15	[19]

by overviewing IoT deployments, with the purpose of selecting some relevant use cases that we use as reference points in our evaluation (Section II). Next, we introduce the reference ICN architecture, discussing aspects related to naming, security, and forwarding (Section III). We then formally state our problem and outline the methodology, introducing the energy model for ICN-IoT deployments at a high level (Section IV). We incrementally add details to the overall picture, refining each of the building blocks (Section V–VI). The full details of the model, which we use to quantitatively and qualitatively contrast geographic and named-based ICN forwarding for the identified use cases (Section VII) are presented next. Finally, we discuss and summarize our main findings (Section VIII).

II. REFERENCE IOT DEPLOYMENTS

Depending on the use cases and applications, IoT deployments cover a broad spectrum of characteristics in terms of node density, number of nodes, typical topology, traffic patterns, etc. Given that our main aim is to identify under which circumstances, if any, geographic forwarding is more advantageous than classical name-based forwarding, we need to select a number of *specific* use cases, representative of different application *classes*. For the purpose of *quantitative* assessment, we need each use case to precisely report characteristics that are clearly specific to a single deployment. At the same time, provided that the selection is made among carefully defined application classes, we expect that the results for the selected example in any given application class *qualitatively* applies to other deployments in the same class.

To define such classes, we consider the IETF Routing Over Low power and Lossy networks (ROLL) working group[1], which identifies four main use cases: (i) urban sensing [20], (ii) industrial sensing [21], (iii) home automation [22], and (iv) building automation [23]. While not considered by the ROLL working group, environmental sensor networks and machine-to-machine deployments are another class of deployments largely covered in the literature [5], [24]. Without loss of generality, and to give the reader several reference points, we consider four deployments, whose relevant characteristics we summarize in Table I, as well as their classes with respect to the aforementioned ROLL categories. For each deployment, we review the reference documentation or available data to determine the number of neighbours for each node, which we use as input data to our model. The deployments, ordered by increasing node degree in Table I, are:

(A) **Place de la Nation**: as an example of the urban sensing class, we take the Cisco-Paris deployment, which is a joint venture between Cisco, the City of Paris and several start-up companies [16]. The deployment is used to measure and track car and pedestrian traffic and pollution patterns on a highly frequented square in Paris. It consists of 19 cameras, 14 noise sensors, 5 pollution-reading sensors, and 12 wireless access points that report information about user connections. The measured data is open-source and available online [25].

(B) **Great Duck Island**: the Great Duck Island deployment [17] is an environmental sensors network consisting of 150 devices. The sensors were used to observe the habitat of seabirds on an island off the coast of Maine and its evolution with respect to weather conditions.

(C) **CASAS**: as an example of the home automation class, we select CASAS [18], which is a so-called “smart-home in a box”: a ready-to-deploy sensor network that allows any consumer to transform their home in a connected (or "smart") home. It consists of 30 nodes communicating over the 802.15.4 radio channel, including temperature sensors and infrared motion/light sensors.

(D) **Sensor Andrews**: as an example of the building automation class, we select Sensor Andrew [19], a sensor network deployment at Carnegie Mellon University (CMU). More than 1000 devices spread all over the CMU campus report numerous measurements such as electricity consumption or temperature.

III. REFERENCE INFORMATION-CENTRIC THINGS (ICN-IOT) ARCHITECTURE

While it has been shown that ICN is a good fit for IoT [1], [3], [11], [26], a number of aspects important for IoT are missing in the original ICN architectures. Specifically, the main building blocks of the reference ICN-IoT architecture [10] that we need to outline are the following: (i) a *neighbour discovery* and association protocol (Section III-A), which ensures that only trusted nodes are authorized to send packets on the network; (ii) a *secure beaconing* (Section III-B) protocol to handle topology and location changes; (iii) a *forwarding scheme* (Section III-C), to ensure correct forwarding of Interest packets over the network independently of the forwarding algorithm class (i.e., geographic or name-based). We point out that while the main focus of this paper is on (iii), however, (i) and (ii) are instrumental and necessary for correct ICN-IoT operations, and must be accounted for in our evaluation. We thus summarize here the relevant points of the ICN-IoT architecture presented in our prior work [10]. Let us note that our study leaves for future work one important feature of ICN: in-network caching. As memory is a scarce resource on IoT platforms, a better understanding of the opportunities for in-network caching in the ICN-IoT would require a thorough study of current ICN-IoT stacks and their memory usage. We provide a first step in that direction with the results of Section VII-B.

[1] https://datatracker.ietf.org/wg/roll/

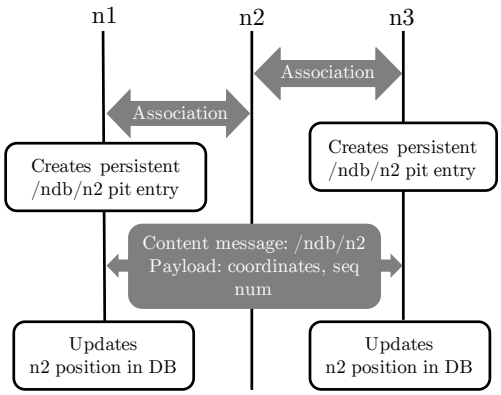


Fig. 1. Synoptic of the ICN-IoT beaconing protocol

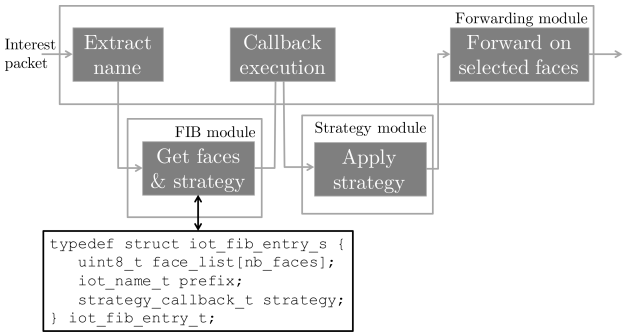


Fig. 2. Coexistence of forwarding strategies in the reference ICN-IoT stack

A. Secure neighbour discovery

In order to protect the network against intruders, sensors must be able to authenticate each other. We initially considered two protocols, with similar security features, based respectively on symmetric [12] and asymmetric [13] cryptography. The study in [13] reveals an interesting trade-off between the communication and the processing/energy overhead: indeed, while the asymmetric-keys based approach incurs a lower traffic overhead (of about 30%), its implementation is significantly more energy- and time-consuming due to the cost of cryptographic operations. Specifically, asymmetric cryptography requires up to 41× more energy and 8× more time for both old (TelosB) and new (OpenMote) generation of IoT platforms.

Therefore, we choose symmetric cryptography for ICN-IoT neighbour discovery, and particularly, we take the state-of-the-art OnboardICNg [12] protocol as a reference. In summary, an OnboardICNg exchange allows two nodes to verify that both have been registered to a trusted third party. As a result, OnboardICNg provides the nodes with a shared symmetric key and includes the distribution of a shared broadcast key in each node's neighbourhood (the broadcast key is a symmetric key propagated by one node to its direct physical neighbour to enable encrypted L2 broadcasts). Given that the primary focus of this paper is the evaluation of the energy cost of name-based forwarding and geographical forwarding for ICN-IoT we refer the reader to OnboardICNg [12] for more details.

B. Secure beaconing

Beaconing presents two new challenges. First, *unsecure beaconing* opens the possibility of wormhole or DoS attacks through exhausting the neighbour database or overloading the central processing unit (CPU). Second, beaconing is essentially a *push* operation, which contrasts with the ICN *pull* model.

Security. In order to prevent these threats, sensors must be able to distinguish between beacons originating from trusted and malicious entities. We thus use the broadcast keys provided by OnboardICNg [12] to encrypt our beacons and authenticate their origin. All the subsequent messages are encrypted with the node broadcast key and contain a message authentication code (MAC). Using this encryption, ICN-IoT devices are resistant to flooding attacks from non-authorized nodes. Indeed, only beacons encrypted with the broadcast key of authenticated neighbours are considered, and the corresponding key can only be accessed by trusted nodes. However, the scheme is not resistant to trusted nodes that have been physically tampered with.

Note that if the AES operations have to be performed in software, attackers can send packets with bogus encryption to perform a simple DoS attack against a node's CPU. However, we point out that recent platforms such as the OpenMote section IV-A) are equipped with hardware modules that can perform AES computation. Therefore, these systems compute and check MAC at low CPU and energy cost.

Push. To accommodate the push nature of beacons with ICN, we must slightly modify the specification of ICN exchanges, similar to the work presented in [27]. Specifically, we use *persistent Pending Interest Table (PIT) entries* (i.e., entries that are not purged after being satisfied once) and *unsolicited Data messages* (i.e., Data messages that are emitted without a corresponding Interest message). We describe the beaconing protocol with the help of Figure 1:

(i) After an OnboardICNg association, each node creates a persistent PIT entry (e.g., with a soft timeout) for `/ndb/neigh_id`, where `neigh_id` is the id of the neighbour with whom the exchange was performed.
(ii) Regularly, each node sends a broadcast unsolicited Data packet (encrypted with the node broadcast key) for `/ndb/node_id` containing the beacon information (e.g., the node's coordinates) and a sequence number (to avoid replay attacks).
(iii) Unsolicited Data packets are forwarded to the beacon processing application, thanks to the persistent PIT entry.

Persistent PIT entries and unsolicited messages have a network utilisation advantage over the traditional ICN Interest/Data exchange. Indeed, the traditional scheme requires four packets per pair of neighbour nodes (two exchanges, one per node), so a total of $4Nd$ where N is the total number of nodes and d the average number of neighbours per node. Instead,

with our scheme, each beacon is broadcast to all of the nodes in the neighbourhood, so that only N packets are required.

C. Forwarding

Our reference ICN-IoT architecture is conceived as a framework to perform name-based and geographic forwarding in the ICN-based IoT and is thus independent of the actual variation of geographic forwarding chosen. We achieve this with the workflow summarized in Figure 2. In our ICN-IoT implementation, FIB entries match with faces and strategies. Faces can be either physical neighbours, application or virtual faces (such as the broadcast face). A strategy is a *callback* on the faces in the FIB, that can, for instance, select a face amongst the available ones with a specific metric. For instance, one could use a specific prefix (such as `/g/`) to forward packets through geographic forwarding by linking it to the corresponding strategy in the FIB. Interest packets destined to any other prefix would still be forwarded by name.

While our architecture allows for a variety of forwarding strategies, for quantitative performance evaluation we need to select specific geographic and name-based forwarding strategies that are implemented and executed on real IoT hardware. Routing and forwarding in the IoT world has been the subject of extensive research; we refer the reader to [28] for a taxonomy and survey of algorithms. To guide our selection, we remark that this taxonomy, which categorizes forwarding into flat/hierarchical or location-based strategies, also applies in an ICN-flavoured IoT deployment, where the choice of using flat/hierarchical or location-based *naming schemes* directly maps the choice of a forwarding strategy as well [26].

Location-based. To select our candidate location-based strategy, we remark that most geographic forwarding techniques are based on greedy forwarding (i.e., select the neighbour closest to the destination as a next hop) with either a beacon-based [14], [29] or beacon-less [30], [31] approach. Greedy choices are complemented by recovery techniques to route around sinkholes, as in GPSR [14] or GOAFR+ [29].

The applicability of geographic forwarding to ICN has explored in a limited way, primarily as applied to Vehicle-to-Vehicle (V2V) networks [4], [32], [33] and are designed to exploit V2V characteristics: highly dynamic, fast moving nodes with no battery/CPU constraints that receive long streams of video/audio data. In [34], the authors propose an ICN-IoT routing scheme based on geographic coordinates. However, their proposal assumes a tree-like topology and does not account for potential sinkholes. We additionally note that, despite the availability of many geographic forwarding algorithms in the literature, there are few implementations; e.g., even the most basic and best-known approaches, such as GPSR [14], are not available in modern IoT toolboxes [15] such as Contiki [35] or RIOT [36]. As a representative of location-based strategies, we implemented GPSR [14], a classic and well-understood strategy based on a geographic greedy forwarding algorithm.

In addition to destination coordinates (that are part of a name under the `/g/` prefix), GPSR also requires additional information for the forwarding. Indeed, to avoid local maxima

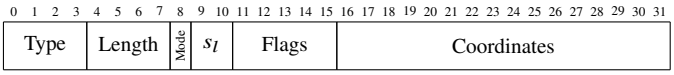


Fig. 3. The TLV field used by GPSR for keeping in-flight state (with $s_l = 16$).

(cases where the current node is closer to the destination than any of its neighbours), GPSR uses a technique called "perimeter routing", which requires the packet to carry the coordinates of the node where it entered the perimeter mode. The ICN-IoT architecture stores this information in a Type Length Value (TLV) field as described in Figure 3, where a flag determines whether the GPSR is in greedy or perimeter mode. Given that we expect the reference ICN-IoT architecture to be used in different scenarios (dense deployment in urban buildings as opposed to sparse deployments in large rural areas), it may be desirable to provide the capability to adjust the coordinate resolution to a specific application scenario to avoid overhead. As a result, the architecture also supports the use of different resolutions for geographic coordinates: as Figure 3 outlines, 2 bits in the flags, noted as s_l, allow to specify from 8 to 64 bits coordinates, in step of 8 bits.

Name-based. In, the matching between Interest names and output faces in the ICN-IoT FIB is usually done using longest-prefix match. Simple flood-and-learn (F&L) forwarding strategy [1], [37] (and variants [38], [39]) are typically used in ICN-IoT to construct FIBs on demand. These are inherently non-scalable and possibly a bad fit for IoT deployments. As such, while we do implement F&L for reference purposes, we do also argue for the need to complementing it with additional techniques to reduce broadcast storms. Particularly, we opt for a Multi-Point Relay (MPR) [40] variant inspired by the Optimized Link State Routing Protocol (OLSR), where at each hop during the message propagation, only a few relays are selected out of those having received the message.

However, while implementing naive F&L is simpler than implementing GPSR, implementing a full-blown MPR distributed OSLR-like protocol is not. In particular, F&L and MPR performance will not only differ in the number of messages sent over the network (which we can simulate) but will also differ in the computational complexity (which we should measure from an actual implementation). We can approximate the computational cost of MPR with the computational cost of the simpler F&L strategy to provide a conservative evaluation of the energy efficiency of MPR for comparison with GPSR. Therefore, we decouple the implementation and prototype only the simpler F&L for the purpose of measuring the computational cost in Section IV-C, and simulate an ideal MPR for the purpose of gathering a lower bound of the number of messages transmitted by MPR over a network in Section VI-B. In particular, our ideal MPR implementation exploits global knowledge available in simulation to find a minimal set of MPR relays, providing a lower bound on the message complexity. These choices guarantee a conservative evaluation of the potential benefits of geographic GPSR over name-based MPR.

TABLE II
CHARACTERISTICS OF THE OPENMOTE

Architecture	ARM Cortex-M3 (32 bits)
MCU	Texas Instrument CC2538 (32MHz)
RAM (ROM)	32KB (512KB)
Encryption HW	AES & ECC
Encryption cost [41]	19.7 μJ (SW, AES-CCM, 128bits) 8.7 μJ (HW, AES-CCM, 128bits)
Consumption [42]	39mW (CPU at 32MHz, no RX/TX) 60mW (CPU idle, RX at -50 dBm) 72mW (CPU idle, TX at 0 dBm)

IV. METHODOLOGY OVERVIEW

In this section, we define the lines along which we evaluate the costs of using name-based and geographic forwarding strategies in ICN-IoT deployments. In particular, we detail the experimental setup (Section IV-A) and focus our attention on the most relevant implementation criteria: namely, (i) *memory footprint* (Section IV-B), (ii) *CPU overhead* (Section IV-C), and (iii) *energy consumption* (Section IV-D). We consider memory and CPU metrics as tied to the *feasibility* of an ICN-IoT deployment, whereas we use the energy consumption to quantify the *cost* of ICN-IoT operation. We found that assessing memory and CPU costs is significantly simpler than assessing the overall energy budget, which, therefore, is the main contribution of our investigation. We thus set to solve the following problem: *under which conditions (if any) is geographic forwarding more performant in terms of energy, memory, and CPU consumption than flood-based strategies for the ICN-IoT?*

A complex system such as an IoT deployment is influenced by numerous factors (that are recapped and summarized further on in Table IV). Therefore, we employ a range of methodologies. At each step of our evaluation, we select the most appropriate one to estimate values that have practical relevance for the different variables. This section defines our ICN-IoT experimental model and enumerates the various sources of energy consumption. Sections V and VI combine measurement from an actual ICN-IoT implementation with stochastic modelling of L2 transmission and simulation of network-wide scenarios to populate the different components of the model, including security, forwarding, data plane traffic, and control plane overhead. The refined model is then quantitatively analyzed to provide guidelines on the most favourable ICN-IoT settings for the different IoT deployment classes and scenarios (Section VII). We point out that, while our previous work [10] accurately assessed the cost of sending *a single ICN packet from a single IoT device*, in this paper we provide a much more complete picture that encompasses *all network-related activities of a full ICN-IoT deployment*.

A. Experimental setup

While our methodology to evaluate the cost of secure geographic forwarding in the reference ICN-IoT architecture is general, the quantitative aspects reported in this paper are relevant for the hardware and software setup with which we conducted our evaluation. To make the quantitative aspects of our evaluation of interest to the largest possible audience, we use the widely used open-source hardware (OpenMote [43]) and software (RIOT OS [36]) stacks.

TABLE III
SIZE OF INTEREST (I), GEO-INTEREST (GI), AND DATA (D) ICN PACKETS

			Packet Type		
	Field	**Field size**	**I**	**GI**	**D**
L2 header	802.15.4 PHY header	6B	✓	✓	✓
	802.15.4 MAC header	23B	✓	✓	✓
	802.15.4 SEC header	5B	✓	✓	✓
L3 header	Packet Type TL	1B	✓	✓	✓
	Nonce TLV	1B (TL) + 1B (V)	✓	✓	
	Name TL	1B	✓	✓	✓
	Name component TLVs	s_n	✓	✓	✓
	GPSR TLV	1B (TL) + (1 + s_l) (V)		✓	
Payload	Content TLV	1B (TL) + s_c (V)			✓
	Signature Info TL	1B			✓
	Signature Type TLV	1B (TL) + 1B (V)			✓
	KeyLocator TLV	1B (TL) + 1B (V)			✓
	KeyId TLV	1B (TL) + 1B (V)			✓
	Signature TLV	1B (TL) + 16B (V)			✓
Footer	802.15.4 Signature	16B	✓	✓	✓
	802.15.4 CRC	2B	✓	✓	✓

Total size	**Packet Type**
56B + s_n	Interest
58B + s_n + s_l	Geo-Interest
79B + s_n + s_c	Data

Hardware setup. We use an OpenMote platform with a 32MHz ARM Cortex-M3 CPU, which is equipped with an IEEE 802.15.4 chipset as well as hardware modules for symmetric and asymmetric cryptography. To evaluate the cost of hardware cryptography module and of receiving or transmitting packets through the IEEE 802.15.4 interface, we rely on measurements performed by Shafagh et al. [41]. The energy consumption figures for this platform are provided in the corresponding datasheet [42], which we summarize along with other characteristics in Table II.

Software setup. Our code runs on top of the RIOT operating system [36]. We implement a custom ICN stack on top of RIOT that uses standard ICN forwarding (i.e., longest-prefix match in the FIB) as well as GPSR (with perimeter routing as introduced earlier). We point out that our code is for the time being closed source, though plans are to make it available in the long term. To accommodate the typically low frame sizes of IoT networks (e.g., 127 bytes for IEEE 802.15.4 networks), adaptations to the TLV-based format of ICN packets have been proposed. Following the recommendations in [44], we implement 1+0 TLVs (i.e., where the Type and Length field are encoded in one single byte), instead of the 1+1 or 2+2 format described in the CCNx specifications [45], with which our implementation is otherwise fully compliant. Table III details the different fields of IEEE 802.15.4 ICN Interest, geographic-Interest and Data frames, as well as reports the total size (as a function of the name or location size) that is instrumental to the model. As shown by this table, the geographic-Interest packet format differs from the ICN Interest only by the presence of the GPSR TLV that is used on top of the name and name-components TLVs to perform the routing.

B. Memory

Memory is a primary constraint in IoT. For example, an old platform such as the MSP430-based TelosB only offers 10KB of random access memory (RAM) and 48KB of flash memory. Even recent hardware like the OpenMote includes only 32KB of RAM and 512KB of flash memory. This amount is still tiny considering that recent implementations of an IoT-ICN stack require already between 5KB [1] and 11KB [46] of RAM. Additionally, optimizing memory consumption is especially interesting in the context of ICN, where caching can be used to accommodate nodes with low duty-cycles [47].

When considering memory requirements, geographic forwarding has advantages over name-based forwarding. Indeed, under GPSR the size of the state retained by a node to be able to forward any packet is bounded by the node degree, whereas under F&L (and variants) each node needs to retain some state for the name of every other node. In fixed ICN networks, state explosion is alleviated by using prefix aggregation in the FIB. However, this is hardly possible in highly-dynamic and mobile IoT networks, and to the best of our knowledge no aggregation scheme for IoT deployments has been proposed for ICN-IoT

Memory requirements can be computed considering that under geographic forwarding, the FIB contains the coordinates (having size s_l) of all its d neighbours, and that the beaconing protocol described in Section III-B additionally requires nodes to store a persistent PIT entry (having size s_{pit}) for each of their neighbours. Under classic name-based forwarding, ICN requires having one FIB entry (having size s_{fib}) for each of the n_s reachable names. As no aggregation scheme is currently available for dynamic IoT topologies, n_s is equal to the number of nodes in the IoT network. This means that each node has a FIB entry for every other node in the network. To compute the required memory, it must be noted that both FIB and PIT entries also contain a 1-byte pointer to an ICN face and that a FIB entry also contains a 1-byte pointer to a strategy. We can thus express the respective memory requirements as:

$$\begin{aligned} M_{geo} &= d(s_l + s_{pit}) = d(s_l + s_n + 1) \\ M_{fib} &= n_s \times s_{fib} = n_s(s_n + 2) \end{aligned} \tag{1}$$

C. Computation

CPU power is another strong constraint on IoT platforms: this holds for both old platforms such as TelosB (16-bit CPU clocked at 8 Mhz), as well as for newer platforms such as the OpenMote (32-bit CPU clocked at 32MHz). An inefficient forwarding algorithm on a slow processor can delay message forwarding, causing congestion in the network.

We remark that the CPU complexity of a forwarding algorithm is also inherently dependent on the underlying hardware: for instance, multiplication on 32-bits integers is much faster on newer 32-bit CPUs than older 16-bit ones. It is thus only possible to evaluate the strength of a forwarding implementation with respect to a specific platform. More specifically, we can evaluate the number of CPU cycles $n_c(algo)$ required in a given assembly language for a specific implementation of any given strategy and then compute the corresponding energy

Listing 1. Benchmarking code

```
uint32_t do_iteration () {
        //Initializes structures and counter
        do_initialize ();
        DWT->CYCCNT = 0;

        //Performs the micro-benchmark
        perform_bench ();

        //returns the number of used CPU cycles
        return DWT->CYCNT;
}
```

consumed by the CPU given its frequency f_{CPU} and its power drain P_{CPU} from data-sheets:

$$\begin{aligned} E_{CPU}(algo) &= P_{CPU}\, t_{CPU}(algo) \\ &= P_{CPU}\, \frac{n_c(algo)}{f_{CPU}} \end{aligned} \tag{2}$$

To devise an accurate model, we need a method to reliably measure the number of cycles $n_c(algo)$. Given that CPU emulators or static code analysis are subject to low accuracy [48], we opt for *micro-benchmarking* the different pieces of the reference ICN-IoT architecture code with cycle-level accuracy, using a simple yet powerful technique. To accomplish micro-benchmarking, we use a special register of the Cortex-M3 CPU dedicated to counting CPU cycles[2]. This register is directly mapped in memory and can be accessed on RIOT through the `DWT->CYCCNT` variable, without performance penalty. An example of the micro-benchmark code reads as presented in listing 1.

D. Energy

The energy cost of a specific ICN-IoT implementation comes from three main sources: a *computational cost* related to the IoT algorithms, a *security cost* related to cryptographic operations and message exchanges due to the security protocol, and a *network cost* related to point-to-point communication, end-to-end transmission, and network maintenance.

During the overall lifetime of an ICN-IoT deployment, network cost can be split into bootstrap, forwarding of Interest/Data packets, and handling of route failures, all of which are clearly dependent on the forwarding strategy employed. Network bootstrap is the cost of setting up the forwarding for the full network to be able to forward packets from any node to any other. Once routes are set up, communication under different forwarding strategies incurs different costs. Indeed, the amount of energy spent for forwarding is related to the computational cost of the forwarding algorithm, as well as to communication costs because of additional state embedded in the Interest packets, and the different numbers of relays under each algorithm. Finally, handling route failure is an operation common to volatile environments such as IoT deployments, where routes to content can become unavailable due to mobility or poor channel conditions. Reacting to this failure triggers the (re)discovery of a path, a costly operation

[2]The CYCCNT register, see http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ddi0337e/ch11s05s01.html

that must be taken into account and whose cost depends on the forwarding strategy.

In our evaluation, we assume that network load is uniformly distributed over the ICN-IoT network, i.e., that each node forwards probabilistically the same amount of traffic. This simplifying assumption, necessary for the tractability of the model, is justified by the fact that we consider networks with either *user/sensor mobility* or *machine-to-machine communications*, which tend to make the existence of a single hotspot sink less prevalent. Furthermore, we do not aim at precise absolute evaluation of the network energy spending in a specific scenario but at a relative comparison of name-based and geographic forwarding for the ICN-IoT. We can then compute the total network cost by summing the $E_{bootstrap}$, $E_{forwarding}$ and E_{change} network-related components as follows:

$$\begin{aligned} E_{total} &= E_{bootstrap} + N_m E_{forwarding} + \frac{N_m}{f_c} E_{change} \\ &= N_m E_{forwarding} + (1 + \frac{N_m}{f_c}) E_{change} \end{aligned} \tag{3}$$

where N_m is the number of useful ICN-IoT queries (i.e., expression of Interest satisfied by a Data packet in a multi-hop fashion) and f_c is the number of queries performed between two route changes. Intuitively, f_c represents the level of dynamism in the network, which lumps altogether phenomena such as (i) to the addition/removal of new names or sensors, (ii) mobility of physical devices, (iii) route failure due to the wireless medium. Arguably, $E_{bootstrap}$ is a one-time cost with vanishing impact over time, so we approximate it with $E_{booststrap} \approx E_{change}$, i.e., the control-plane cost of repopulating a FIB. $E_{forwarding}$ instead represents the energy cost of a single query in the IoT network once the FIB is already populated, and as such accumulates the data-plane costs to transmit an Interest over several hops in the network, as well as the cost of receiving the corresponding Data packet travelling in the opposite direction. $E_{forwarding}$ and E_{change} are addressed in Section V and Section VI respectively.

Finally, it must be noted that the number of messages N_m is not a design parameter. Rather, we can re-express Equation (3) to infer the total number of exchanges N_m that are possible, as a function of the network dynamism f_c, under different forwarding strategies:

$$N_m = \frac{E_{total} - E_{change}}{E_{forwarding} + \frac{1}{f_c} E_{change}} \tag{4}$$

Without loss of generality, in Section VII we exploit (4) where we equate the total energy budget to the amount of energy available in standard AA batteries, i.e., $E_{total} = E_{AA}$.

V. Cost of forwarding a single ICN packet

In this section, we set out to evaluate the energy spent by a node to forward a single ICN packet as a first refinement of our energy model. We then evaluate this model for the OpenMote, using our own experiments and data gathered from the literature. From the point of view of a relay, the packet forwarding process can be divided into 5 steps, namely: (i) frame reception, (ii) frame decryption, (iii) forwarding face selection through the forwarding strategy, (iv) frame encryption, and (v) frame transmission. We summarize the various variables used in the mode in Table IV.

TABLE IV
Summary of variables used in the evaluation

Parameter	Symbol	Default value
No. of neighbours	d	15
No. of ICN names	n_s	2000
No. of FIB entries	n_f	15
Size of a location info	s_l	8 bytes
Size of a name	s_n	$\lceil \log_2(n_s) \rceil$
Size of the content	s_c	32 bytes
Energy cost of AES encryption	E_{AES}	10 μJ
No. of tries / transmission	$n_{tr,s}$	eq. (6)
L2 drop probability	p_c	[49]
No. of times a packet is forwarded during flood (including L2 retries)	$N_{tr}(T, D)$	eq. (21)
Energy cost of transmission/bit	E^b_{tx}	1.163 μJ
Energy cost of reception/bit	E^b_{rx}	0.96 μJ
Max number of hops of a packet on the WSN	T	8
Size of an ICN Interest packet	s_i	56B + s_n
Size of a geographic ICN packet	$s_{i,g}$	58B+s_n+s_l
No. of Interest/Content exchanges before a route change	f_c	free parameter
Energy content of an AA battery	E_{AA}	15390J
Budget of Interest/Data exchanges during the lifetime	N_m	eq. (20) and eq. (19)

A. Frame transmission and reception

The transmission and reception cost of an ICN packet is given by the amount of time that the node's antenna has to be powered in transmission (TX) and reception (RX) mode. Since the power consumption P_{tx} (resp. P_{rx}) of the platform is dependent on the hardware and available from the data sheets provided by the manufacturer, only the transmission time needs to be computed.

At any hop, the transmission time is driven by two factors: the number of retransmissions that are necessary for a successful reception on the wireless medium, and the size of the message (which impacts the time taken by each retransmission). Let $n_{tr,s}(p_c)$ be the average number of tries necessary for a successful transmission and C_{phy} the channel capacity with a collision probability p_c. The transmission cost of a frame of size s_f (the reception cost can be similarly derived) is then given by:

$$E_{tx}(s_f) = P_{tx} \frac{s_f}{C_{phy}} n_{tr,s}(p_c) = E^b_{tx} s_f n_{tr,s}(p_c) \tag{5}$$

Let M_{tr} be the maximum number of L2 retransmissions of a given frame (after which the frame is dropped). For a given p_c, we have that $n_{tr,s}(p_c)$ is, in expectation:

$$\begin{aligned} \mathbb{E}[n_{tr,s}(p_c)] &= \sum_{k=1}^{M_{tr}} k P(k \text{ L2 transmissions needed}) \\ &= \sum_{k=1}^{M_{tr}} k p_c^{k-1}(1 - p_c) \\ &= \frac{1 - p_c^{M_{tr}}(M_{tr} + 1) + p_c^{M_{tr}+1} M_{tr}}{1 - p_c} \end{aligned} \tag{6}$$

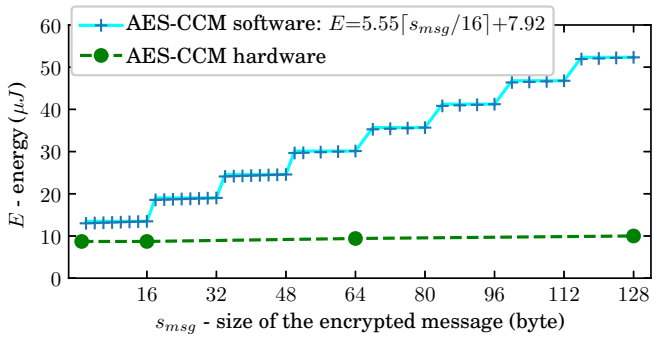


Fig. 4. Energy cost of AES-CCM based on size of message. Software encryption measurement are gathered as described in Section IV-C. Measurement for hardware encryption are provided in Table 6 of [41]

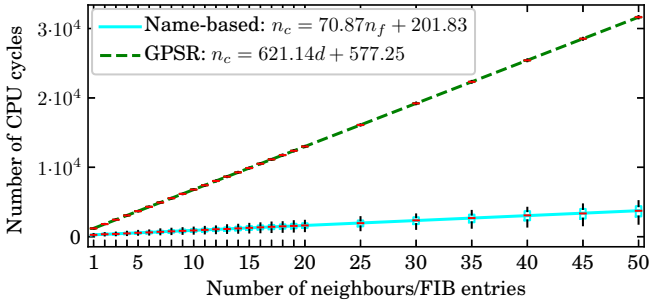


Fig. 5. Cycles per forwarding decision on the OpenMote based on number of neighbours/FIB entries. Boxplots report cycle-level accurate measurements with the method in Section IV-C, lines reports linear fitting of the data.

We stress that p_c is not a system parameter but depends in turn on other properties of the ICN-IoT deployment, such as node density and radio range. We come back to p_c in Section VI.

B. Data Encryption and Decryption

The cost of cryptography depends on the device's capabilities, such as CPU characteristics, and more importantly on the availability of hardware cryptography components. We present the energy consumption of AES-CCM encryption in the OpenMote platform, considering both software and hardware-assisted cryptography as a function of the message's size in Figure 4.

For the software implementation, we microbenchmark the AES implementation of the `crypto` module of RIOT with the previously outlined technique. Figure 4 provides the measurements as well as an equation derived from the measurements. For the hardware-assisted encryption, we point out that cryptography hardware modules are not yet supported on RIOT: we thus use as a reference the AES-CCM measurement reported in Table 6 of [41]. The picture clearly shows the importance of hardware modules for cryptographic operations: the AES-CCM software implementation consumes up to 5 times more energy than its hardware-assisted counterpart. In hardware, AES-CCM has a maximum cost of $10\mu J$ per packet (since the IEEE 802.15.4 maximum transmission unit (MTU) is 127B).

In the rest of the paper, we assume that the platform is equipped with a hardware module that is accessible through an API of the software stack (as planned in RIOT). With hardware-assisted encryption, as Figure 4 shows, we can assume that costs for encryption/decryption are constant with respect to the frame size. Finally, given that encryption/decryption costs are not the main component of a packet transmission, for the sake of simplicity, we assume that encryption and decryption have an equal cost, which we indicate with E_{AES}.

C. Forwarding algorithm

Deducing the cost of the forwarding algorithm is fairly simple: Equation (2) requires a microbenchmark of the forwarding code to accurately measure the number of CPU cycles required to perform either name-based ICN forwarding (i.e., longest-prefix match in the ICN FIB) or GPSR forwarding operations. Note that in this section, we do not yet consider the control traffic, but that in this context the name-based FIB would typically be populated using F&L or MPR. These measurements are reported in Figure 5, which shows boxplots of the number of CPU cycles as a function of the number of entries in the FIB (for name-based ICN) or the number of neighbours of the node (for GPSR) in the x-axis (incidentally, the figure shows a linear correlation between CPU consumption and memory occupancy irrespectively of the forwarding algorithm). As expected, geographic forwarding grows more steeply than name-based ICN forwarding (621 cycles per additional neighbour versus 71 cycles per additional FIB entry). Indeed, geographic forwarding requires the node to perform floating point multiplications to compute the distance to the next hops, while ICN forwarding consists only of byte comparisons. It must be noted that we could implement geographic forwarding using fixed-point arithmetic or a more recent CPU with embedded support for floating-point operations. However, while this would narrow the performance gap, geographic forwarding would still be more expensive in CPU cycles than the byte comparisons used by name-based forwarding.

While the gap between geographic and name-based forwarding appears to be important, it is just one component of the overall cost, which we detail in the next section.

D. Overall cost

We can now express the total cost for a node to relay an ICN packet as:

$$E_{relay}(algo, s) = E_{tx}(s) + E_{rx}(s) + 2E_{AES} + P_{CPU}\frac{n_{cycles(algo)}}{f_{CPU}} \quad (7)$$

It should be noted that since Data packets are forwarded through symmetric routing, they are not concerned by the computation overhead. If we take into account both the Interest and the Data packet, the forwarding cost per node adds up to:

$$E_{relay}(algo) = E_{relay}(algo, s_i) + E_{relay}(\text{exact-match}, s_c) \quad (8)$$

Finally, we can summarize $E_{forwarding}$ as the cost of forwarding an Interest packet and its corresponding Data

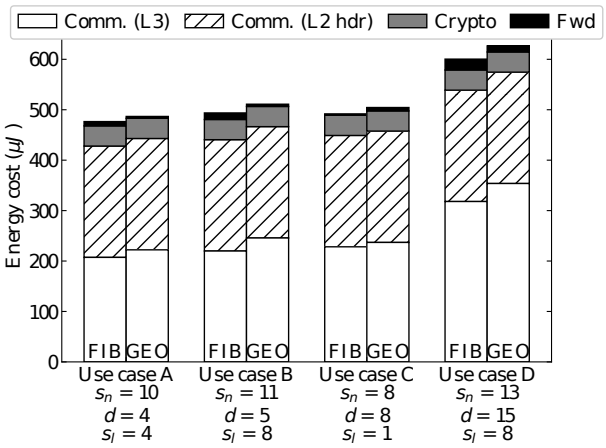


Fig. 6. Energy cost of the forwarding modules

packet over a multi-hop network by considering that on the path we have: (i) $T-1$ relay nodes that must perform both TX and RX operations, (ii) a source node that only performs TX for the Interest and RX for the Data packet, and (iii) a destination that performs only RX for the Interest and TX for the Data packet. This can be rewritten as T relay nodes performing both TX and RX operations for both packets:

$$E_{forwarding} = T(E_{relay}(algo, s_i) + E_{relay}(\text{exact-match}, s_c)) \quad (9)$$

In Figure 6, we represent the respective costs of each of the different components for the reference use cases in Table I with the corresponding numbers of FIB entries, neighbours and the size of the name. In particular, we group transmission and reception costs, and present them as L2 IEEE 802.15.4 and L3 ICN components, on which the cryptographic and forwarding costs are stacked. For each reference use case, we gather d and n_s from the literature, then compute $s_n = \log_2(n_s)$, and finally select a size of s_l relevant to the geographic scale of the use case (i.e., depending on the coordinate resolution).

From Figure 6, it clearly emerges that RX and TX operations are the predominant factors of energy consumption (which is underestimated as we do not account for MAC layer signalling nor idle listening [50]). The communication cost is two orders of magnitude higher than the cost of forwarding (software), even for geographic forwarding with numerous neighbours, and one order of magnitude higher than the cost of cryptography (hardware). Notice that the cost of software forwarding lumps altogether energy expenditures related to CPU (selecting the forwarding face) and memory (storing and updating the neighbour table or the FIB), which are clearly negligible w.r.t. the energy spent on cryptographic and network operations.

Hence, to summarize:

- the principal overhead in energy consumption when using GPSR is the *increased header size* included in each Interest packet because of the GPSR TLV,
- the complexity of the forwarding algorithm is *clearly negligible* and Equation (7) becomes:

$$E_{relay}(algo, s) \approx E_{tx}(s) + E_{rx}(s) + 2E_{AES} \quad (10)$$

TABLE V
ADDITIONAL BYTES SENT IN THE REFERENCE ICN-IOT ARCHITECTURE

Beacons (per node)	Interest packet (per packet & hop)
$58 + s_{id} + s_l$	$2 + s_l$

VI. COST OF CONTROL TRAFFIC

An additional source of energy consumption is the background control information that is needed to discover routes to new names, new neighbours, and to maintain the network connectivity in spite of changes such as node mobility. That cost is intrinsically related to the geographic (Section VI-A) or flooding-based (Section VI-B) forwarding algorithm employed. In this section, we thus build a model to derive the energy consumption of the network forwarding process (data and control plane). We complete this model with simulation to estimate the spread of flooding for the F&L and MPR algorithms.

A. Geographic forwarding

For geographic forwarding, control information takes two forms: (i) the beacons to transmit geographic information between neighbours, and (ii) the additional GPSR TLV in the Interest packet to transmit per-packet forwarding state along the path. We now review the cost of both these factors, which we summarize in Table V.

Beacons. Beacons are the most obvious source of control overhead in geographic forwarding. As described in section III-B, they are local (i.e., not routed) broadcast messages that do not propagate in the network and only reach the nodes involved by the L2 broadcast.

Let us consider a node whose immediate neighbourhood has changed: this node must broadcast its current position to its new neighbours, and receive the d broadcast messages from its neighbours. Given s_b the size of a beacon message, the energy E_{change} required for this update by each node is simply:

$$E^{GPSR}_{change} = (s_b n_{tr,s} E^b_{tx} + E_{AES}) + d(s_b n_{tr,s} E^b_{rx} + E_{AES}) \quad (11)$$

Headers. On top of the beacons, additional control information for GPSR is embedded in every Interest packet through the TLV described in Section III-C. As shown in Figure 3, this TLV contains both a flag for the forwarding mode (greedy or perimeter) and a set of coordinates. Thus, the size $s_{i,g}$ of an Interest packet for geographic forwarding is:

$$s_{i,g} = s_i + 2 + s_l \quad (12)$$

As we have seen in Figure 6, we expect the extra control fields in the headers to have a sizeable impact on data traffic, lowering the efficiency of GPSR with respect to name-based forwarding. We can now plug $s_{i,g}$ in Equation (9) and using Equation (10), we get:

$$E^{GPSR}_{forwarding} = T\left(4E_{AES} + n_{tr,s}(s_{i,g} + s_c)(E^b_{rx} + E^b_{tx})\right) \quad (13)$$

At the same time, we expect the benefits of GPSR to come primarily from keeping the amount of FIB state bound to the

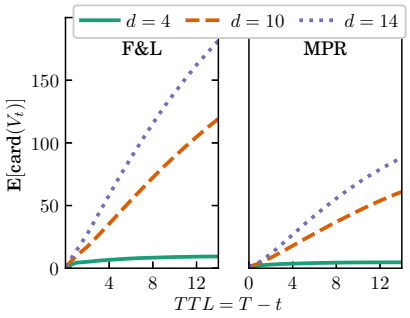


Fig. 7. Average value of card(V_t)

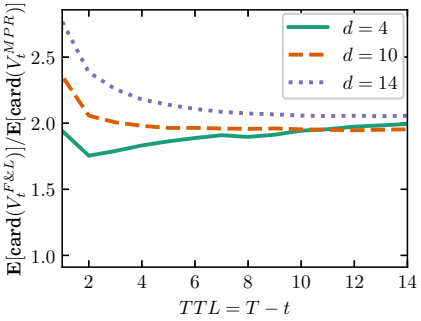


Fig. 8. Ratio between average values of card(V_t) for naive F&L and MPR

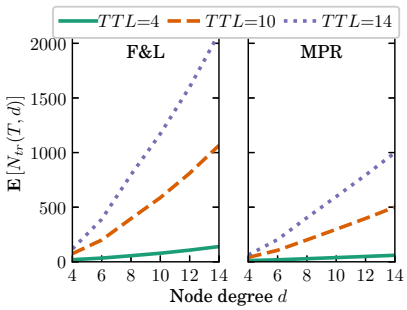


Fig. 9. Expected number of transmitted messages $\mathbf{E}[N_{tr}(T,d)]$

number of neighbours and limiting the exchanges at a local level, unlike F&L-based strategies.

B. Flood and learn

In case of name-based forwarding, despite similar forwarding operations, the data-plane energy is lower due to the smaller Interest size ($s_i < s_{i,g}$):

$$E^{F\&L}_{forwarding} = T\left(4E_{AES} + n_{tr,s}(s_i + s_c)(E^b_{rx} + E^b_{tx})\right) \quad (14)$$

However, we need to the estimate the cost of learning routes to objects in an F&L-based ICN-IoT deployment:

$$E^{F\&L}_{change} = \frac{N_{tr}(T,d)}{n_{tr,s}} 2E_{AES} + N_{tr}(T,d)(E^b_{rx} + E^b_{tx})s_i \quad (15)$$

Therefore, we have to determine the number of network-wide transmissions required to update FIB information $N_{tr}(T,d)$, for which we model the propagation of a flooded packet over a wireless multihop network. Let us consider a uniform random geographic graph $G(N,d)$ where N is the number of nodes in G and d the average node degree, and let focus on a message m that must be flooded over the IoT network with a maximum Time-To-Live (TTL) of T. Denoting with $N_{tr}(T,d)$ the number of times m has been transmitted during its propagation, including L2 retransmissions, we have:

$$\mathbf{E}[N_{tr}(d)] = \sum_{t=0}^{T-1} \mathbf{E}[n^t_{tr}(d)] \quad (16)$$

where $n^t_{tr}(d)$ is the number of times the message m is transmitted at the t-th hop (i.e., for a TTL=$T-t$). Letting V_t be the set of nodes that relay message m at hop t, and recalling that $n_{tr,s}(m)$ is the number of transmission attempts until a successful transmission, we have:

$$n^t_{tr}(d) = \sum_{i \in V_t} n_{tr,s}(p_c) \quad (17)$$

and thus:

$$\mathbf{E}[n^t_{tr}(d)] = \mathbf{E}[\text{card}(V_t)]\mathbf{E}[n_{tr,s}(p_c)] \quad (18)$$

To estimate the number of nodes card(V_t) transmitting the message at the t-th hop, we simulate the packet propagation for both F&L and MPR. For this purpose, we have developed multi-threaded programs that we have made available to the community[3]. The tools generate random graphs with a given density and number of nodes and use a custom version of breadth-first search to compute card(V_t) in the case of naive F&L. In the case of MPR, they compute the set of MPR-neighbours using the greedy algorithm described in [40]. For any given density, we sample a population of 10^5 random graphs on which we evaluate the number of transmission for naïve F&L and MPR from a random source. We run the simulations on a Linux 4.7 server with an Intel Xeon CPU clocked at 2.40GHz: for each density, the simulation takes about 9 hours, where the dominant[4] time is represented by the MPR strategy. As simulation time is rather long, we also provide on our GitHub the results of our simulation rounds as well as a Jupyter Notebook to explore them.

Figure 7 presents simulation results for the number of messages generated by naïve F&L (left) and MPR (right) as a function of $T - t \in [0, 15]$ (x-axis) and for different density values $d \in [4, 15]$. To quantify the advantages brought by MPR, Figure 8 additionally reports the ratio of the messages generated by F&L over MPR. Interestingly, regardless of the density, MPR roughly halves the number of messages that need to be flooded at each step. To perform a conservative assessment of the benefits of geographic forwarding, we thus only consider MPR as a benchmark.

Finally, plugging in Equation (18) the card(V_t) measured in simulations, we can accurately numerically estimate the forwarding cost of flood-based strategies:

$$\mathbf{E}[N_{tr}(d)] = \sum_{t=0}^{T-1} \mathbf{E}[\text{card}(V_t)] \frac{1 - p_c(d)^{M_{tr}}(M_{tr}+1) + p_c(d)^{M_{tr}+1} M_{tr}}{1 - p_c(d)} \quad (21)$$

where the collision probability $p_c(d)$ for IEEE 802.15.4 is given in [49] as a function of the average node degree (that is referred as p_{netcol} in [49]). Results for Equation (21) are reported in Figure 9.

VII. Guidelines for ICN-IoT operation

Using these results, we can systematically compare MPR and geographic forwarding for both energy (Section VII-A)

[3] https://github.com/marceleng/geographic-icthings

[4] While Python is enough for F&L, we had to rewrite the tool in C for speed efficiency in the case of MPR. Both tools are available in the GitHub page

$$N_m^{MPR} = \frac{E_{AA} - \left(\frac{N_{tr}(T,d)}{n_{tr,s}} 2E_{AES} + N_{tr}(T,d) s_i (E_{rx}^b + E_{tx}^b)\right)}{\left(\frac{N_{tr}(T,d)}{n_{tr,s}} 2E_{AES} + N_{tr}(T,d) s_i (E_{rx}^b + E_{tx}^b)\right) / f_c + T\left(4E_{AES} + n_{tr,s}(s_i + s_c)(E_{rx}^b + E_{tx}^b)\right)} \quad (19)$$

$$N_m^{GPSR} = \frac{E_{AA} - \left((s_b n_{tr,s} E_{tx}^b + E_{AES}) + d(s_b n_{tr,s} E_{rx}^b + E_{AES})\right)}{\left((s_b n_{tr,s} E_{tx}^b + E_{AES}) + d(s_b n_{tr,s} E_{rx}^b + E_{AES})\right) / f_c + T\left(4E_{AES} + n_{tr,s}(s_{i,g} + s_c)(E_{rx}^b + E_{tx}^b)\right)} \quad (20)$$

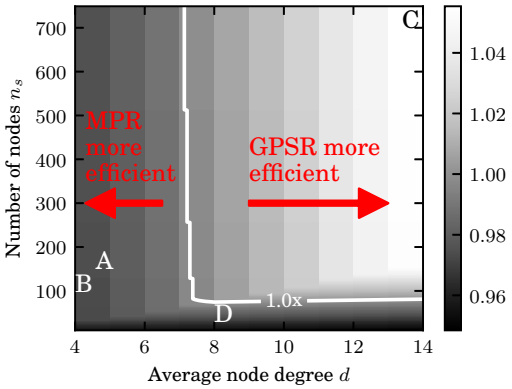


Fig. 10. Relative message budget $\frac{N_m^{GPSR}}{N_m^{MPR}}$ when $f_c = 60$

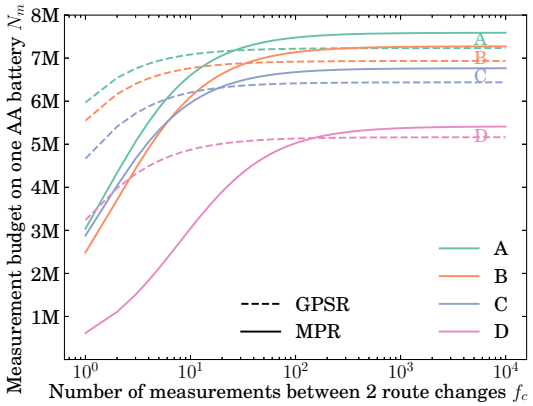


Fig. 11. Expected message budget for GPSR and MPR for the reference deployments as a function of network dynamism

and feasibility (Section VII-B) metrics. In particular, we apply the models derived in Section V and Section VI to the four IoT reference deployments of Section II, considering a network of OpenMote devices. Readers or ICN-IoT operators interested in other network characteristics are referred to our Jupyter Notebook[5], whose interactive interface can be used to explore more scenarios. For convenience, we summarize the main parameters in the evaluation in Table IV, as well as their default scalar or functional values when relevant.

A. Energy cost

Specifically, having modelled the energy budget required to transmit N_m exchanges, we can derive the number of Interest/Data exchanges for name-based forwarding ($N_m^{F\&L}$) and geographic forwarding (N_m^{GPSR}) that can be completed with the energy available in one AA battery $E_{AA} = 15$ kJ [51]. For completeness, the message budgets are reported in Equation (19) and Equation (20).

These equations provide a metric for measuring the efficiency of both protocols: the number of possible Interest/Content exchanges (N_m) for a given battery capacity, depending on the density d, the average number of hops T and the topology change ratio f_c. Figure 10 shows the ratio between N_m^{GPSR}/N_m^{MPR} (i.e., the relative message budget of GPSR vs MPR) depending on the network size n_{nodes} (y-axis) and the average node degree d (x-axis). In this figure, the lighter the heatmap, the more efficient geographic forwarding is with respect to MPR. Without loss of generality, we select $T = 8$ and $f_c = 60$, which corresponds to the case where, if an Interest/context exchange happens every minute, then a route change happens every 1 hour.

Several conclusions can be drawn from Figure 10. First, let us note that the ratio takes values $N_m^{GPSR}/N_m^{MPR} \in [0.95, 1.05]$, i.e., the respective performances of name-based and geographic forwarding are close for these (T, f_c) settings. The inflexion point in the figure is caused by a change in the MPR regime: i.e., at some point, the network becomes big enough so that the flooding stops because of the TTL, rather than because everyone in the network already received the packet. Also, note that in these settings, 3 (out of 4) reference deployments (A, B, D) sit below the $N_m^{GPSR} = N_m^{MPR}$ contour and would thus (slightly) benefit from using MPR. Finally, it is easy to see that for an average node degree higher than 7 and a deployment size larger than 100 nodes, geographic forwarding performs (slightly) better than MPR.

We next turn our attention to dynamic cases where we vary the rate at which the network changes (e.g., due to node churn, mobility, deployment of new nodes, etc.). Specifically, f_c represents the number of consecutive data plane exchanges between two changes requiring control plane messages. We let $f_c \in [1, 10^4]$, so that for $f_c = 1$ there is a significant control plane overhead, whereas for $f_c = 10^4$ the network is mostly stable. We report the raw number of Interest/Data messages on an AA battery as a function of f_c for the four reference deployments on Figure 11. Two main observations hold. First, for networks with frequent changes geographic forwarding is up to twice as efficient as MPR for all considered scenarios. Second, it can be seen that MPR is slightly more efficient than geographic forwarding over relatively stable networks, although the difference is small enough not to have practical

[5]https://github.com/marceleng/geographic-icthings/blob/master/models/geographic-icthings.ipynb

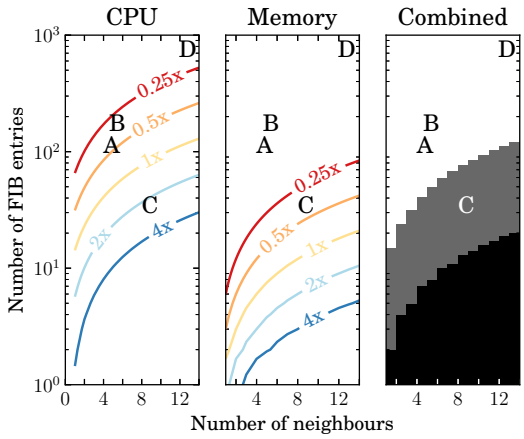


Fig. 12. Contours of the relative memory and CPU footprints of GPSR and F&L ICN. The heatmap shows areas where GPSR outperforms F&L ICN for both criteria (white), for memory utilisation (grey) or for neither (black)

relevance. Thus, GPSR seems a good candidate for dynamic scenarios, while both GPSR and MPR can be used in stable topologies at approximately the same cost.

B. Memory and CPU complexity

Finally, we concisely summarize the CPU and memory footprint of geographic forwarding and F&L-based strategies in Figure 12. The contour plots show the relative footprint of GPSR versus F&L ICN, while the heatmap illuminates areas where GPSR is advantageous on both criteria (white), only in memory (grey) or in neither (black). The letters show the respective positions of the use cases described in Table I. The picture shows that GPSR has a lower memory footprint when the number of FIB entries inflates, which is an important factor on memory-constrained nodes in networks where numerous names must be accessible. In particular, three of the use cases (A, B, and D) require less memory and CPU resources using GPSR rather than with name-based forwarding. Furthermore, while CPU consumption is often favourable to F&L ICN, GPSR is faster in sparse but large networks (e.g., $n_f > 40$ and $d < 5$). Overall, there are no CPU/memory obstacles to implement geographic forwarding in ICN-IoT. Rather, GPSR can yield to memory savings with respect to name-based forwarding, an appealing advantage for constrained environments.

VIII. Conclusion

In this paper, we introduce an ICN-IoT architecture, which is a generic framework capable of performing secure geographic forwarding over ICN in IoT deployments. We use this framework to assess the CPU/memory feasibility of geographic-based forwarding, as well as to derive energy models for geographic and name-based forwarding, that encompass all network-related activities of an IoT deployment. We implement GPSR in a RIOT-based ICN stack and contrast it to naïve flooding, as well as to an improved flooding technique using Multi-Point Relay (MPR).

In summary, we find that GPSR-based forwarding is feasible in ICN-IoT. Specifically, GPSR memory requirements are lower than that of flood-based strategies. Additionally, while algorithmic complexity increases when using GPSR over flooding-based strategies, the required CPU resources are a negligible component of the overall energy cost. Indeed, the cost of security (including cryptographic operations and network overhead) is at least one order of magnitude higher than the computation cost of the forwarding algorithm, which can, therefore, be neglected.

In terms of energy consumption, two opposite forces are in play: the increased header size in the case of GPSR translates into higher energy cost per unit packet while GPSR keeps beaconing local, lowering the cost of network-wide updates. As such while there is a clear incentive in using GPSR for dynamic networks requiring frequent updates, the performance gap in the case of networks with stable topologies and infrequent changes is slightly favourable to flood-based strategies. At the same time, that gap remains small and it should not be a limiting factor for the use of GPSR in ICN-IoT deployments.

Acknowledgements

This work has been carried out at LINCS (http://www.lincs.fr) and benefited from the support of NewNet@Paris, Cisco's Chair "Networks for the Future" at Telecom ParisTech (https://newnet.telecom-paristech.fr)

References

[1] E. Baccelli, et al. "Information centric networking in the IoT: Experiments with NDN in the wild." In *Proc. 1st Int. Conf. Information-centric Networking*, ACM, New York, NY, USA, pp. 77–86, 2014.

[2] M. Amadeo, et al. "Information-centric networking for the internet of things: challenges and opportunities." *IEEE Network*, vol. 30, no. 2, pp. 92–100, March 2016.

[3] Y. Zhang, et al. "Requirements and challenges for IoT over ICN." Internet-Draft draft-zhang-icnrg-icniot-requirements-01, Internet Engineering Task Force, Apr 2016. Work in Progress.

[4] G. Grassi, et al. "Navigo: Interest forwarding by geolocations in vehicular named data networking." In *IEEE 16th Int. Symp. World of Wireless, Mobile and Multimedia Networks*, Boston, MA, USA, Jun 2015.

[5] M. Amadeo, et al. "Information-centric networking for M2M communications: Design and deployment." *Comput. Commun.*, vol. 89, pp. 105 – 116, 2016.

[6] M. Chen. "NDNC-BAN: supporting rich media healthcare services via named data networking in cloud-assisted wireless body area networks." *Inform. Sci.s*, vol. 284, pp. 142–156, 2014.

[7] H. Yue, et al. "Dataclouds: Enabling community-based data-centric services over the internet of things." *IEEE Internet of Things J.*, vol. 1, no. 5, pp. 472–482, Oct 2014.

[8] W. Shang, et al. "Securing building management systems using named data networking." *IEEE Network*, vol. 28, no. 3, pp. 50–56, 2014.

[9] M. Amadeo, et al. "Information centric networking in IoT scenarios: The case of a smart home." In *2015 IEEE Int. Conf. Commun.*, pp. 648–653, June 2015.

[10] M. Enguehard, et al. "SLICT: Secure localized information centric things." In *ACM ICN Workshop Inform. Centric Things for 5G*, ACM, New York, NY, USA, pp. 255–260, 2016.

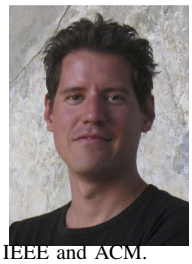
[11] J. Burke, et al. "Secure sensing over named data networking." In *IEEE 13th Int. Symp. Network Computing and Applicat.*, pp. 175–180, Aug 2014.

[12] A. Compagno, et al. "OnboardICNg: a secure protocol for on-boarding IoT devices in ICN." In *Proc. 3rd ACM Conf. Information-Centric Networking*, ACM, New York, NY, USA, pp. 166–175, 2016.

[13] M. Enguehard, et al. "Poster: On the cost of secure association of information centric things." In *Proc. 3rd ACM Conf. Information-Centric Networking*, ACM, New York, NY, USA, pp. 207–208, 2016.
[14] B. Karp and H. T. Kung. "GPSR: Greedy perimeter stateless routing for wireless networks." In *Proc. 6th Annu. Int. Conference Mobile Computing and Networking*, ACM, New York, NY, USA, pp. 243–254, 2000.
[15] O. Hahm, et al. "Operating systems for low-end devices in the internet of things: A survey." *IEEE Internet of Things J.*, vol. 3, no. 5, pp. 720–734, Oct 2016.
[16] A. Peters. "Paris is redesigning its major intersections for pedestrians, not cars." *Fast Company*, Aug 2016.
[17] R. Szewczyk, et al. "An analysis of a large scale habitat monitoring application." In *Proc. 2nd Int. Conf. Embedded Networked Sensor Syst.*, ACM, New York, NY, USA, pp. 214–226, 2004.
[18] D. J. Cook, et al. "CASAS: A smart home in a box." *IEEE Comput.*, vol. 46, no. 7, pp. 62–69, July 2013.
[19] A. Rowe, et al. "Sensor andrew: Large-scale campus-wide sensing and actuation." *IBM J. of Research and Develop.*, vol. 55, no. 1.2, pp. 6:1–6:14, Jan 2011.
[20] M. Dohler, et al. "Routing requirements for urban low-power and lossy networks." RFC 5548, RFC Editor, May 2009.
[21] K. Pister, et al. "Industrial routing requirements in low-power and lossy networks." RFC 5673, RFC Editor, Oct 2009.
[22] A. Brandt, et al. "Home automation routing requirements in low-power and lossy networks." RFC 5826, RFC Editor, Apr 2010.
[23] J. Martocci, et al. "Building automation routing requirements in low-power and lossy networks." RFC 5867, RFC Editor, Jun 2010.
[24] P. Corke, et al. "Environmental wireless sensor networks." *Proc. IEEE*, vol. 98, no. 11, pp. 1903–1917, Nov 2010.
[25] Mairie de Paris. "Paris data." https://opendata.paris.fr.
[26] M. Amadeo, et al. "Multi-source data retrieval in IoT via named data networking." In *Proc. 1st Int. Conf. Information-centric Networking*, ACM, New York, NY, USA, pp. 67–76, 2014.
[27] C. Tsilopoulos and G. Xylomenos. "Supporting diverse traffic types in information centric networks." In *Proc. ACM SIGCOMM Workshop Information-centric Networking*, pp. 13–18, 2011.
[28] J. N. Al-Karaki and A. E. Kamal. "Routing techniques in wireless sensor networks: a survey." *IEEE Wireless Commun.*, vol. 11, no. 6, pp. 6–28, Dec 2004.
[29] F. Kuhn, et al. "Geometric ad-hoc routing: Of theory and practice." In *Proc. 22nd Annu. Symp. Principles of Distributed Computing*, ACM, New York, NY, USA, pp. 63–72, 2003.
[30] M. Heissenbüttel, et al. "BLR: beacon-less routing algorithm for mobile ad hoc networks." *Comput. Commun.*, vol. 27, no. 11, pp. 1076 – 1086, 2004. Applications and Services in Wireless Networks.
[31] J. A. Sanchez, et al. "BOSS: Beacon-less on demand strategy for geographic routing in wireless sensor networks." In *2007 IEEE Int. Conf. Mobile Adhoc and Sensor Syst.*, pp. 1–10, Oct 2007.
[32] L. Wang, et al. "MobiCCN: Mobility support with greedy routing in content-centric networks." In *2013 IEEE Global Commun. Conf.*, pp. 2069–2075, Dec 2013.
[33] D. Pesavento, et al. "A naming scheme to represent geographic areas in NDN." In *IFIP Wireless Days*, pp. 1–3, Nov 2013.
[34] H. Ma, et al. "On networking of internet of things: Explorations and challenges." *IEEE Internet of Things J.*, vol. 3, no. 4, pp. 441–452, Aug 2016.
[35] A. Dunkels, et al. "Contiki - a lightweight and flexible operating system for tiny networked sensors." In *IEEE 29th Annu. Conf. Local Comput. Networks*, pp. 455–462, Nov 2004.
[36] O. Hahm, et al. "RIOT OS: Towards an OS for the internet of things." In *Proc. 32nd IEEE Int. Conf. Comput. Commun.*, Apr 2013.
[37] M. A. Hail, et al. "Caching in named data networking for the wireless internet of things." In *2015 Int. Conf. Recent Advances in Internet of Things*, IEEE, pp. 1–6, 2015.
[38] C. Anastasiades, et al. "Dynamic unicast: Information-centric multi-hop routing for mobile ad-hoc networks." *Comput. Networks*, vol. 107, pp. 208–219, 2016.
[39] M. Amadeo, et al. "Named data networking for IoT: An architectural perspective." In *2014 European Conf. Networks and Commun. (EuCNC)*, pp. 1–5, June 2014.
[40] A. Qayyum, et al. "Multipoint relaying for flooding broadcast messages in mobile wireless networks." In *Proc. 35th Annu. Hawaii Int. Conf. Syst. Sci.*, pp. 3866–3875, Jan 2002.
[41] H. Shafagh, et al. "Talos: Encrypted query processing for the internet of things." In *Proc. 13th ACM Conf. Embedded Networked Sensor Syst.*, ACM, New York, NY, USA, pp. 197–210, 2015.
[42] Texas Instrument. *CC2538 Powerful Wireless Microcontroller System-On-Chip for 2.4-GHz IEEE 802.15.4, 6LoWPAN, and ZigBee® Applications*, Dec 2012. Revised April 2015.
[43] X. Vilajosana, et al. "OpenMote: open-source prototyping platform for the industrial IoT." In *Int. Conf. Ad Hoc Networks*, Springer, pp. 211–222, 2015.
[44] M. Mosko and C. Tschudin. "CCN and NDN TLV encodings in 802.15.4 packets." https://www.ietf.org/mail-archive/web/icnrg/current/pdfs9ieLPWcJI.pdf, 2015. Consulted on March 17, 2017.
[45] PARC. "The CCNx project." https://blogs.parc.com/ccnx/.
[46] W. Shang, et al. "The design and implementation of the NDN protocol stack for RIOT-OS." Tech. rep., Technical Report NDN-0043, NDN, 2016.
[47] O. Hahm, et al. "A named data network approach to energy efficiency in IoT." In *IEEE GLOBECOM Workshop Named Data Networking for Challenged Commun. Environments*, 2016.
[48] K. Roussel, et al. "Using Cooja for WSN simulations: Some new uses and limits." In *Proc. 2016 Int. Conf. Embedded Wireless Syst. and Networks*, Junction Publishing, USA, pp. 319–324, 2016.
[49] S. Pollin, et al. "Performance analysis of slotted carrier sense IEEE 802.15.4 medium access layer." *IEEE Trans. Wireless Commun.*, vol. 7, no. 9, pp. 3359–3371, September 2008.
[50] G. De Meulenaer, et al. "On the energy cost of communication and cryptography in wireless sensor networks." In *IEEE 4th Int. Conf. Wireless and Mobile Computing, Networking and Commun.*, IEEE, pp. 580–585, 2008.
[51] K. Nisimova. "Energy of a 1.5 V battery." http://hypertextbook.com/facts/2001/KhalidaNisimova.shtml. Consulted on 20 April 2017.

Marcel Enguehard (S'17) received the M.Sc. degree jointly from Ecole Polytechnique, Palaiseau, France, and KTH - Royal Institute of Technology, Stockholm, Sweden in 2016. He is currently working toward the Ph.D. degree at Telecom ParisTech, Paris, France, conjointly with Cisco Systems, Issy les Moulineaux, France. His research focuses on forwarding mechanism for Information-Centric networking applied to the Internet of Things, spanning from sensor networks to efficient data processing in multi-tiered IoT platforms.

Ralph E. Droms received his B.Sc. and M.Sc. degrees from Pennsylvania State University, and his Ph.D. degree from Purdue University. He is currently Staff Software Engineer at Google. He has contributed extensively to the IETF, authoring more than 25 RFCs, including many of the core DHCP specifications, as well as more than 20 journal and conference papers. Dr. Droms chaired the IETF DHC working group until 2009, then served as an Internet Area Director in the IESG and as a member of the IAB.



Dario Rossi (S'03–M'05–SM'13) received his M.Sc. and Ph.D. degrees from Politecnico di Torino in 2001 and 2005 respectively, was a visiting researcher at University of California, Berkeley during 2003-2004, and is currently Professor at Telecom ParisTech and Ecole Polytechnique. He has coauthored over 150 conference and journal papers, received 4 best paper awards, a Google Faculty Research Award (2015) and an IRTF Applied Network Research Prize (2016). He is a Senior Member of IEEE and ACM.