SLICT: Secure Localized Information Centric Things

Marcel Enguehard
Cisco Systems &
Telecom ParisTech
mengueha@cisco.com

Ralph Droms
Cisco Systems
rdroms@cisco.com

Dario Rossi
Telecom ParisTech
dario.rossi@telecom-paristech.fr

ABSTRACT

While the potential advantages of geographic forwarding in wireless sensor networks (WSN) have been demonstrated for a while now, research in applying Information Centric Networking (ICN) has only gained momentum in the last few years. In this paper, we bridge these two worlds by proposing an ICN-compliant and secure implementation of geographic forwarding for ICN. We implement as a proof of concept the Greedy Perimeter Stateless Routing (GPSR) algorithm and compare its performance to that of vanilla ICN forwarding. We also evaluate the cost of security in 802.15.4 networks in terms of energy, memory and CPU footprint. We show that in sparse but large networks, GPSR outperforms vanilla ICN forwarding in both memory footprint and CPU consumption. However, GPSR is more energy intensive because of the cost of communications.

CCS Concepts

•**Networks** → **Routing protocols;** *Sensor networks; Wireless mesh networks;* Security protocols;

Keywords

Information Centric Network; Geographic Forwarding; Wireless Sensor Network

1. INTRODUCTION

Several pieces of work have demonstrated the appeal of ICN for IoT deployments. Bacelli et al. [3], show that an almost out-of-the-box ICN stack (specifically, NDN [24]) outperforms the standard IPv6-based network stack (IEEE 802.15.4, 6LoWPAN and RPL). However, several challenges must be overcome to apply ICN to IoT (see Zhang et al. [25]). *Efficient packet delivery* with *minimal control traffic* is one such challenge. Geographic forwarding, where packets are forwarded towards a location instead of a host, aims to solve it by keeping routing updates local. Indeed, the packet destination is embedded in its network header by the sender and nodes only require their direct neighbours' positions to forward it. Another challenge on power-constrained nodes is *network access control*. Sensor data is often private (e.g., health sensor readings) and must be protected from malicious attackers. Furthermore, if a dishonest node joins a WSN, it can perform denial of service (DoS) attacks by flooding malicious packets and thus draining the forwarding node's battery. In this paper, we jointly address these two challenges, proposing an ICN-IoT design able to transmit sensor data *geographically* and *securely* over the network.

We start our review by considering the choice of a naming strategy in ICN-IoT, a crucial piece that can be divided in two approaches: request-based vs host-based names. In the request-based approach, introduced by Intanagonwiwat et al. [11], names encode user requests (e.g., "temperature at time t in zone Z"). The network must then find a sensor able to fulfil the request. On the other end of the spectrum, explicit host-based names have been proposed [25, 3]. In this case, names are directly linked to the sensor that generates the data (e.g., "temperature at time t as measured by sensor s"), sometimes at the cost of a name resolution system. This approach is required in scenarios (such as healthcare or factory automation) where sensors/actuators must be authenticated, such as those we consider in this paper.

The choice of a name structure may induce modifications to the ICN architecture. For instance, Amadeo et al. [2] suggest changes to the NDN specifications (such as longest-prefix match and entries matching multiple content packets in the PIT) to accommodate scenarios where several sensors are producers for the same name. On the access control side, Burke et al. [4] and Compagno et al. [5] present naming and communication patterns to enforce access control on ICN-based WSN. We build on these premises, with a complementary angle as we assess the cost (in terms of CPU cycles and energy required) of desirable features such as access control on ICN-IoT hardware from a practical viewpoint.

Geographic forwarding allows nodes to forward packets with only local information. While geographic forwarding has been largely studied in wireless networks in general [1], its application in IoT is still deficient: there is no GPSR [12] implementation in Contiki [6] or RIOT [9]. The applicability of geographic forwarding to ICN has inspired a few pieces of work, especially targeting Vehicular area networks [21, 15, 8], whose characteristics are however different from WSN. Vehicular networks are highly dynamic with fast nodes, which have few battery/CPU constraints and receive long streams of data (e.g., video or audio). In contrast, WSN consists of volatile but mostly static nodes, with short data sequences

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ICN'16, September 26 - 28, 2016, Kyoto, Japan

© 2016 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ISBN 978-1-4503-4467-8/16/09…$15.00

DOI: http://dx.doi.org/10.1145/2984356.2988519

and strict CPU and energy budgets.

Our contributions can be summarized as follows: (i) SLICT, a framework to enable geographic forwarding for ICN-IoT, (ii) an implementation of GPSR for ICN-IoT, (iii) an evaluation setup for forwarding strategies specific to ICN-IoT, that we apply to SLICT and vanilla ICN forwarding. In the remainder of this paper, we overview our proposal (sec. 2), describe our methodology (sec. 3) and results (sec. 4), before outlining our current research directions (sec. 5).

2. SLICT OVERVIEW

In this section, we describe the general functioning of SLICT, the framework we design to perform secure geographic routing over ICN-based WSN, and detail its various components. The main building blocks of SLICT are (i) a *neighbour discovery* and association protocol which ensures that only trusted nodes are authorized to send packets on the network; (ii) a *secure beaconing* protocol to handle topology and location changes; (iii) *naming scheme* to ensure the forwarding of interest packets over the network independently of the geographic forwarding paradigm.

2.1 Neighbour Discovery

In order to protect the network against intruders, sensors must be able to authenticate each other. We initially considered two protocols, based respectively on symmetric (OnboardICNg [5]) and asymmetric [7] cryptography with similar security features. A preliminary investigation, not reported here for lack of space but available in [7] let us observe an interesting trade-off between the communication and the CPU/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 (it requires up to 41× more energy and 8× more time) for both old (Telos B) and new (OpenMote) generation of IoT boards. We thus fix the choice of OnboardICNg for SLICT neighbour discovery.

An OnboardICNg exchange allows two nodes to verify that both have been registered to a trusted third party. It 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.

2.2 Secure beaconing

At the same time, two new challenges arise. First, *unsecure beaconing* opens the possibility of wormhole attacks or Denial of Service attack by neighbourhood database overload. 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 beacons originating from trusted entities from malicious ones. We thus use the broadcast keys provided by OnboardICNg [5] 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).

This way, SLICT is resilient 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 resilient against trusted nodes that have been tampered with).

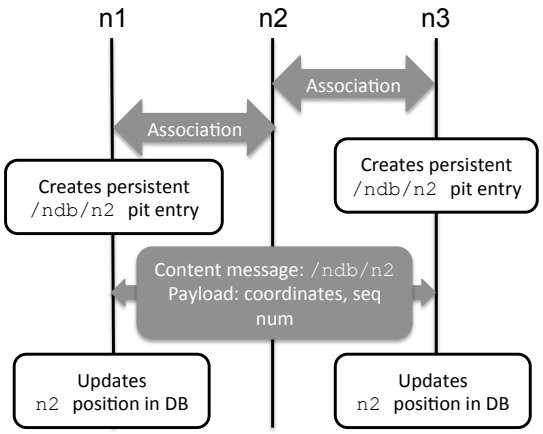

Figure 1: SLICT beaconing protocol

Push. To accommodate the push nature of beacons we ICN, we must slightly bend the specification of ICN exchanges, similarly to the work presented in [20]. More specifically, we use *persistent PIT entries* (i.e., entries that are not purged after being satisfied once) and *unsolicited content messages* (i.e., content messages that are emitted without a corresponding interest message). Our protocol, represented in fig. 1, uses these tools in the following fashion:

(i) After an OnboardICNg exchange, 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 content packet for `/ndb/node_id` containing the beacon information (e.g., the node's coordinates) and a sequence number (to avoid replay attacks). That packet is encrypted with the node's broadcast key.

(iii) These unsolicited content packets are forwarded to the beacon processing application thanks to the persistent node entry.

Persistent PIT entries and unsolicited messages has a network utilisation advantage over the traditional ICN interest/content exchange. First, 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. With our scheme, each beacon is broadcasted to the whole neighbourhood thus only N packets are required.

2.3 Geographic forwarding

Naming. In principle, two naming strategies would support geographic forwarding on the ICN-IoT network, whose main difference is the importance given to the destination geographic coordinates. One option is to consider geographic coordinates only as forwarding "hints" which are only loosely followed, whereas another possibility is to forward strictly according to geographic coordinates. In the former case, coordinates need not to be part of the name, and could be

Bits	8	8	1	2	5	s_{loc}
Field	opcode	length	mode	s_{loc}	flags	coordinates

Figure 2: The GPSR TLV

stored as Type Length Value (TLV) fields in the packet header, which would make it possible to resort to name-based routing on legacy devices. In the latter case, it would be preferable to make coordinates an integral part of a name (e.g., encoded as `/coord`$_x$`coord`$_y$`/rest/of/name`).

In SLICT, we opt for the strict approach and consider destination coordinates as part of the name. However, as we describe in sec. 2.3, we also use a TLV to convey additional information necessary for the forwarding.

GPSR implementation. SLICT is conceived as a framework to perform geographic forwarding in ICN-based WSNs and is thus independent of the actual variation of geographic forwarding chosen. 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 [12, 14] or beacon-less [10, 16, 23] approach. Greedy choices are complemented by recovery techniques to circle around sinkholes (see for instance GPSR [12] or GOAFR+ [14]) whereas more involved approaches require to represent network topology in non-Euclidean spaces (such as the hyperbolic space [13, 22]). As a proof of concept, we implemented GPSR [12], a classic geographic forwarding algorithm based on greedy-forwarding, that is however not available in modern WSN toolboxes (Contiki or RIOT).

To avoid local maxima (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. SLICT stores this information in a TLV as described in fig. 2, where a flag determines whether the GPSR is in greedy vs perimeter mode. SLICT also supports for different resolutions of geographic coordinates (2 bits in the flags, noted as s_{loc}, allow to specify from 8 to 64 bits coordinates, in step of 8 bits). Given that SLICT can be used in different scenarios (dense deployment in urban buildings vs sparse deployments in large rural areas), it may be desirable to leave the application developers the possibility to tune the resolution to the scenario to optimize the SLICT overhead.

3. EXPERIMENTAL SETUP

In terms of methodology, our evaluation is based on a combination of several sources of information: figures provided by the hardware's manufacturer and by previous literature, as well as micro-benchmarks of our ICN software on production hardware. In this section, we detail the hardware and software setup with which we conducted our evaluation.

Hardware setup. We evaluate the cost of secure geographic forwarding in SLICT using an OpenMote board with a 32MHz ARM Cortex-M3 CPU. The OpenMote board is shipped with an IEEE 802.15.4 chipset as well as hardware modules for symmetric and asymmetric cryptography. To evaluate the cost of cryptography and of receiving or transmitting packets through the 802.15.4 interface, we rely on measurements performed by Shafagh et al. [18]. The energy

Table 1: Characteristics of the OpenMote board

Architecture	ARM Cortex-M3 (32 bits)
MCU	Texas Instrument CC2538 (32MHz)
RAM (ROM)	32KB (512KB)
Encryption HW	AES & ECC
Encryption cost	19.7 μJ [18] (SW, AES-CCM, 128bits) 8.7 μJ [18] (HW, AES-CCM, 128bits)
Consumption	39mW (CPU at 32MHz, no RX/TX) 60mW (CPU idle, RX at -50 dBm) 72mW (CPU idle, TX at 0 dBm)

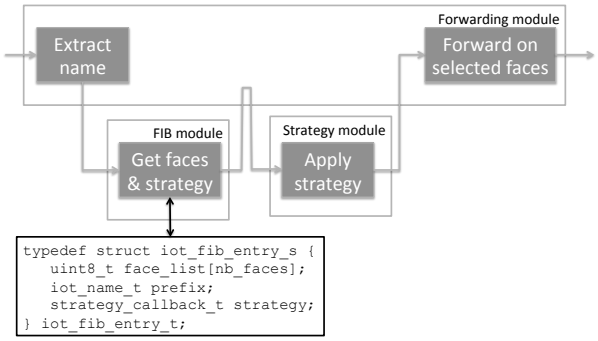

Figure 3: Interest handling in our ICN stack

consumption figures for this board are provided in the corresponding datasheet [19], which we summarize along with other characteristics in table 1.

Software setup. Our code runs on top of the RIOT operating system [9]. We implement a custom ICN stack on top of RIOT that uses standard ICN forwarding (i.e., longest-prefix match in the FIB) and that we plan to make available on the long term. In our 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 allows for instance to select a face amongst the available ones with a specific metric. This workflow is summarized in fig. 3 To perform GPSR, we created an independent library that implements vanilla GPSR forwarding as described in [12] in 220 lines of C code. We then match the `/g/` prefix in our FIB to a virtual face that represents all the physical neighbours and a strategy that links to our GPSR library. To the best of our knowledge, this is the first implementation of GPSR ported on RIOT.

Micro-benchmark. We micro-benchmark the different pieces of SLICT code with cycle-level accuracy, using a simple yet powerful technique. Instead of using CPU emulators or static code analysis, we used a special register of the Cortex-M3 CPU dedicated to counting CPU cycles[1]. This register is directly mapped in memory and can be accessed on RIOT through the `DWT->CYCCNT` variable. Thus, our micro-benchmark code reads as presented in listing 1.

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

Listing 1: Benchmarking code

```
uint32_t do_iteration () {
        //Randomly populate required structures
        prepare_structures ();

        //Reinitialises the cycle counter
        DWT->CYCCNT = 0;

        //Performs the micro-benchmark
        perform_test ();

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

Table 2: Variables used in the evaluation

Parameter	Symbol	Default value
Number of neighbours	n_{neigh}	15
Number of ICN names	n_{names}	2000
Number of FIB entries	n_{fib}	15
Size of a location info	s_{loc}	8 bytes
Size of a name	s_{name}	$\lceil \log_2(n_{names}) \rceil$

4. EXPERIMENTAL RESULTS

In this section, we evaluate the performance of our GPSR implementation and compare it to a vanilla implementation of ICN over IoT, using longest-prefix match in the FIB (sec. 4.1). We then evaluate the cost of security in SLICT (sec. 4.2). In both sections, we focus our attention on the most relevant criteria for WSNs implementation: energy consumption, memory overhead and network utilisation. Clearly, a complex system such as a WSN is influenced by numerous variables: in table 2, we present the different variables that we used during the evaluation, as well as their default values unless otherwise specified.

4.1 The Cost of Geographic Forwarding

We first evaluate the differences between geographic forwarding using GPSR and vanilla ICN forwarding. We estimate the energy consumption of both mechanisms, as well as their memory and network overhead. We also discuss the potential impact of control traffic on these estimations.

4.1.1 Network overhead

There are two different sources of network overhead due to geographic forwarding in SLICT: the beaconing mechanism and the additional TLV described in sec. 2.3. We present the overhead per component in table 3.

If c_t is the cost of a transmitting a byte, c_r the cost of receiving a byte, d_i the average interest-arrival frequency and d_b the beaconing frequency, the average energy cost of the network overhead per node of SLICT depending on the time t is:

$$ov_{net}(t) = (66 + s_{loc})d_b(c_t + dc_r)t + (3 + s_{loc})(c_r + c_t)d_i t$$

4.1.2 Memory and CPU overhead

In a first step, we evaluate the computation performance gap between GPSR and vanilla ICN. Figure 4 shows the CPU cycles required to perform either vanilla ICN forwarding or GPSR forwarding. The x-axis represents respectively the number of entries in the FIB or the number of neighbours

Table 3: Additional bytes sent because of SLICT

Beacons (per node)	Interest packet (per packet & hop)
$66 + s_{loc}$	$3 + s_{loc}$

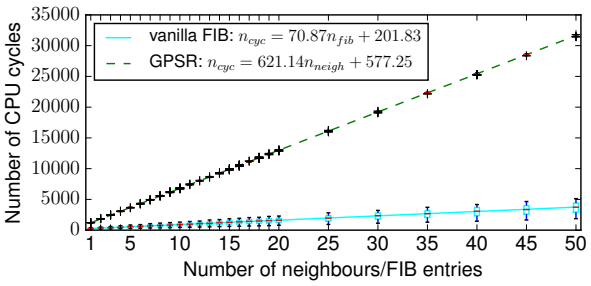

Figure 4: Cycles per forwarding decision on the OpenMote vs number of neighbours/FIB entries

of the node. As expected, geographic forwarding grows more steeply than vanilla 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 vanilla forwarding consists only of byte comparisons.

SLICT requires sensors to retain information about their neighbours. More specifically, each additional neighbour requires a position (i.e., s_{loc} bytes) and a PIT entry for the beaconing protocol. However, since the sensor forwards packets through the coordinates in the name, it does not require any entry in its FIB. Thus, the differential in memory utilisation between geographic forwarding and vanilla ICN forwarding can be computed by:

$$\begin{aligned}\Delta_{mem} &= n_{neigh}(s_{loc} + s_{pit}) - n_{fib} \times s_{fib} \\ &= n_{neigh}(s_{loc} + s_{name} + 1) - n_{fib}(s_{name} + 2)\end{aligned} \quad (1)$$

We summarize the CPU and memory footprint of geographic forwarding w.r.t. vanilla ICN in fig. 5. The contour plots show the relative footprint of GPSR versus vanilla ICN, while the heatmap illuminates areas where GPSR is more performant on both criteria (white), only in memory (grey) or in neither (black). It 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. Furthermore, while CPU consumption is often favourable to vanilla ICN, GPSR is faster in sparse but large networks (e.g., $n_{fib} > 40$ and $n_{neigh} < 5$).

4.1.3 Total energy consumption

We use the linear fittings in fig. 4 to estimate the energy cost of forwarding a packet for a sensor in an IoT network. This cost can be broken down in three modules: (i) the cost of receiving and transmitting the packets through the antenna, (ii) the cost of decrypting and encrypting the packet, (iii) the cost of the forwarding algorithm. In fig. 6, we represent the respective costs of these modules depending on the number of FIB entries, neighbours and the size of the name.

Figure 6 shows that the predominant factor of energy consumption is the RX and TX operations (which are even un-

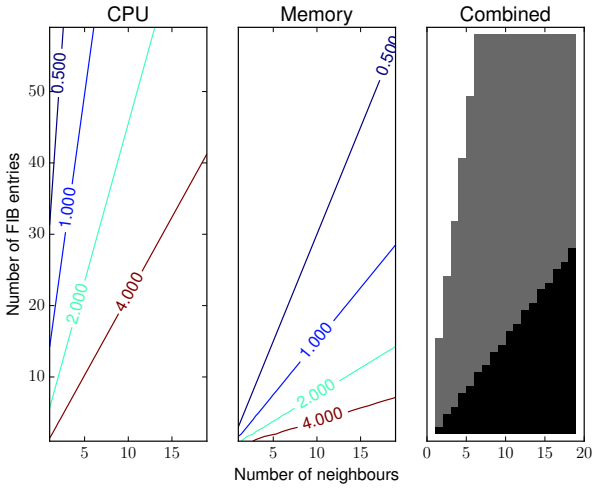

Figure 5: Contours of the relative memory and CPU footprints of GPSR vs vanilla ICN. The heatmap shows areas where GPSR outperforms vanilla ICN for both criteria (white), for memory utilisation (grey) or for neither (black)

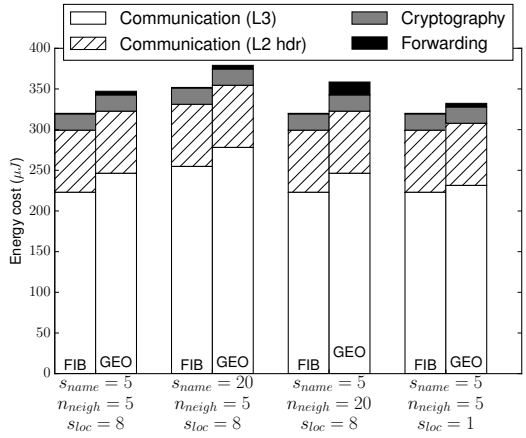

Figure 6: Energy cost of the forwarding modules

dervalued since they do not account for MAC layer signalling and potential retransmissions). The communication cost is two orders of magnitude higher than the cost of forwarding, even for geographic forwarding with numerous neighbours, and one order of magnitude higher than the cost of cryptography. Thus, the principal overhead in energy consumption when using GPSR is the additional bytes included in each interest packet because of the GPSR TLV. Since content packets are forwarded through symmetric routing, they are not concerned by this overhead.

4.2 The Cost of Security

Security in SLICT comes at a price. It is paid during the association process, but also during each transmission since all packets must be decrypted before going through the forwarding module (and then encrypted again before being forwarded to the next hop). We evaluate the cost of security in terms of computation, network and memory footprint.

Table 4: Additional fields in the 802.15.4 frame when using AES-CCM-128

Field	Security header	Frame Counter	Key Control	Encrypted MAC
Size	6-14B	4B	1B	16B

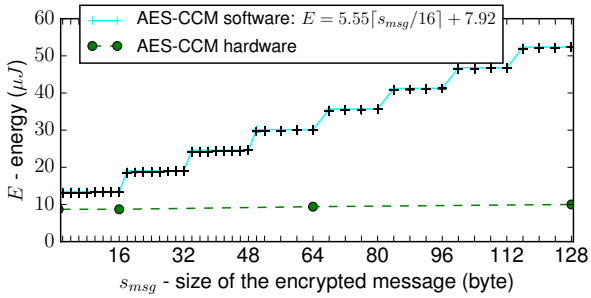

Figure 7: Energy cost of AES-CCM vs size of message. Hardware numbers provided by [18] (table 6)

4.2.1 *Network overhead*

802.15.4 security features. The usage of AES-CCM-128 with 802.15.4 is standardized. It requires additional fields in the L2 and L3 layer that we listed in table 4. The security header appears in the 802.15.4 MAC header to indicate the encryption setup, while the other fields are needed by AES-CCM-128 [17]. Its size depends on the size of the source key identifiers (for instance, a 6-byte Ethernet address). The total overhead per 802.15.4 frame is thus between 28 and 35 bytes per frame (e.g., on every beacon, interest or content messages).

OnboardICNg. According to [5], each OnboardICNg association requires a total of 835 bytes exchanged on the network (518 for the joining node and 317 for the other). Using the notations from sec. 4.1.1, the energy cost averages to $835 \times \frac{c_t + c_r}{2} \times n_{neigh}$ per node.

4.2.2 *Memory and computation overhead*

The computation overhead occurs from the messages decryption and encryption necessary on both ends of the forwarding process. The cost of cryptography is variable from one device to another. It depends on the CPU as well as on the availability of hardware cryptography components. We present the energy consumption of AES-CCM encryption in both hardware and software depending on the message's size in fig. 7. The software implementation is the AES implementation of the `crypto` module of RIOT. As hardware encryption is not supported yet on RIOT, we used the numbers from Table 6 of [18] for hardware AES-CCM. It shows the importance of hardware modules for cryptographic operations: AES-CCM consumes up to 5 times more energy in software than hardware. In hardware, AES-CCM has maximum cost of $10\mu J$ per packet (since the 802.15.4 MTU is 127B). As shown in fig. 6, this is one order of magnitude lower than the cost of receiving or transmitting the messages.

The memory footprint of security is mostly due to retaining keying material. In SLICT, a node must retain its own broadcast key and two keys per associated neighbour (the

Table 5: Overhead of security in μJ per component

OnboardICNg	Computation	Network
$715 \times n_{neigh}$	≈ 20/packet	$(59 - 74)$/packet

shared key and the broadcast key). This translates to a total of $16 + 32n_{neigh}$ bytes. Even in large networks (>20 neigbhours/node), this is negligible compared to the total memory available on recent boards such as the OpenMote.

4.2.3 Total energy consumption

We present in table 5 the overhead of security depending on the source in SLICT. It shows that again network overhead is the major factor of energy consumption. Furthermore, it also shows that security is almost one order of magnitude more expensive than GPSR in terms of energy consumption. While GPSR vs FIB forwarding is a choice that is non-necessarily critical, many applications require hop-by-hop encryption and access control. The major source of overhead in SLICT is thus non-negotiable.

5. DISCUSSION

Our evaluation of the cost of geographic forwarding is only preliminary. First of all, while we mention the cost associated with our beaconing protocol, we do not take into account the cost of updating FIB entries in the vanilla ICN case. Indeed, an important property of geographic forwarding is the locality of control traffic (only neighbours must learn about topology changes), compared to protocols like RPL where topology changes must be propagated in the network. Furthermore, a large part of our evaluation is based on estimated values. We intend to directly measure the energy consumption of the OpenMote during the different phases of SLICT thanks to a digital ammeter. Furthermore, we plan to deploy SLICT on a large testbed to measure its network-wide performance (e.g., convergence time, control traffic overhead, request completion latency).

Moreover, GPSR is not necessarily an optimal geographic algorithm choice. It does not guarantee delivery in all planar graphs for instance. Hyperbolic routing [22] guarantees packet delivery at the trade-off of an extended stretch (i.e., packets need more hops to be delivered). Furthermore, hyperbolic routing does not require to embed TLVs in the packet, thus reducing its network overhead. Such approaches deserve to be evaluated in the ICN-IoT.

6. CONCLUSION

In this paper, we presented SLICT, a framework to perform secure geographic forwarding over ICN in IoT deployments. We study the CPU, memory, network and energy footprint of GPSR and show that it outperforms vanilla ICN forwarding in certain scenarios. However, the additional bytes that have to be transmitted with interest packets render GPSR less performant in terms of energy consumption. Nonetheless, this evaluation does not account for certain advantages of geographic forwarding such as localised routing updates or ease of management. We also show that the cost of security (both cryptographic operations and network overhead) is at least one order of magnitude higher than the cost of the forwarding algorithm.

7. REFERENCES

[1] J. N. Al-Karaki and A. E. Kamal. Routing techniques in wireless sensor networks: a survey. *IEEE Wireless Communications*, 11(6), 2004.

[2] M. Amadeo, C. Campolo, et al. Multi-source data retrieval in iot via named data networking. In *ACM ICN '14*.

[3] E. Baccelli, C. Mehlis, et al. Information centric networking in the IoT: Experiments with NDN in the wild. In *ACM ICN '14*.

[4] J. Burke, P. Gasti, et al. Secure sensing over named data networking. In *IEEE NCA' 14*.

[5] A. Compagno, M. Conti, et al. OnboardICNg: a secure protocol for on-boarding IoT devices in ICN. In *ACM ICN '16*. In press.

[6] A. Dunkels, B. Gronvall, et al. Contiki - a lightweight and flexible operating system for tiny networked sensors. In *IEEE LCN 2004*.

[7] M. Enguehard, R. Droms, et al. Poster: On the cost of secure association of Information Centric Things. In *ACM ICN '16*. In press.

[8] G. Grassi, D. Pesavento, et al. Navigo: Interest forwarding by geolocations in vehicular Named Data Networking. In *IEEE WoWMoM 2015*.

[9] O. Hahm, E. Baccelli, et al. RIOT OS: Towards an OS for the internet of things. In *IEEE INFOCOM 2013*.

[10] M. Heissenbüttel, T. Braun, et al. BLR: beacon-less routing algorithm for mobile ad hoc networks. *Computer Communications*, 27, 2004. Applications and Services in Wireless Networks.

[11] C. Intanagonwiwat, R. Govindan, et al. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *ACM MobiCom '00*.

[12] B. Karp and H. T. Kung. GPSR: Greedy perimeter stateless routing for wireless networks. In *ACM MobiCom' 00*.

[13] R. Kleinberg. Geographic routing using hyperbolic space. In *IEEE INFOCOM' 07*.

[14] F. Kuhn, R. Wattenhofer, et al. Geometric ad-hoc routing: Of theory and practice. In *ACM PODC '03*.

[15] D. Pesavento, G. Grassi, et al. A naming scheme to represent geographic areas in NDN. In *Wireless Days (WD), 2013 IFIP*, pages 1–3. 2013.

[16] J. A. Sanchez, R. Marin-Perez, et al. Boss: Beacon-less on demand strategy for geographic routing inwireless sensor networks. In *IEEE MASS 2007*.

[17] N. Sastry and D. Wagner. Security considerations for ieee 802.15.4 networks. In *ACM WiSe '04*.

[18] H. Shafagh, A. Hithnawi, et al. Talos: Encrypted query processing for the internet of things. In *ACM SenSys '15*.

[19] Texas Instrument. *CC2538 Powerful Wireless Microcontroller System-On-Chip for 2.4-GHz IEEE 802.15.4, 6LoWPAN, and ZigBee® Applications*, 2012. Revised April 2015.

[20] C. Tsilopoulos and G. Xylomenos. Supporting diverse traffic types in information centric networks. In *ACM SIGCOMM ICN workshop*. 2011.

[21] L. Wang, O. Waltari, et al. MobiCCN: Mobility support with greedy routing in content-centric networks. In *IEEE GLOBECOM' 13*.

[22] W. Zeng, R. Sarkar, et al. Resilient routing for sensor networks using hyperbolic embedding of universal covering space. In *IEEE INFOCOM' 10*.

[23] H. Zhang and H. Shen. Energy-efficient beaconless geographic routing in wireless sensor networks. *IEEE TPDS*, 21(6), 2010.

[24] L. Zhang, D. Estrin, et al. Named data networking (NDN) project. Technical Report NDN-0001, University of California, Los Angeles, 2010.

[25] Y. Zhang, D. Raychadhuri, et al. Requirements and Challenges for IoT over ICN. Internet-Draft draft-zhang-icnrg-icniot-requirements-01, Internet Engineering Task Force, 2016. Work in Progress.