Designing Time Slotted Channel Hopping and Information-Centric Networking for IoT

Oliver Hahm, Cédric Adjih, Emmanuel Baccelli, Thomas Schmidt, Matthias Wählisch

Designing Time Slotted Channel Hopping and Information-Centric Networking for IoT

Oliver Hahm*, Cédric Adjih*, Emmanuel Baccelli*, Thomas C. Schmidt†, Matthias Wählisch‡
*INRIA †HAW Hamburg ‡Freie Universität Berlin
{oliver.hahm,emmanuel.baccelli,cedric.adjih}@inria.fr, t.schmidt@haw-hamburg.de, m.waehlisch@fu-berlin.de

***Abstract*—Recent proposals to simplify the operation of the IoT include the use of Information Centric Networking (ICN) paradigms. While this is promising, several challenges remain. In this paper, our core contributions (a) leverage ICN communication patterns to dynamically optimize the use of TSCH (Time Slotted Channel Hopping), a wireless link layer technology increasingly popular in the IoT, and (b) make IoT-style routing adaptive to names, resources, and traffic patterns throughout the network—both without cross-layering.**

***Index Terms*—IoT, NDN, TSCH, 802.15.4e, name-based routing, adaptive forwarding**

I. Introduction

The current Internet is based on IP as convergence layer, and focuses primarily on the interconnection between machines—the byproduct of which being that these machines can then store, send and receive digital content. ICN proposes a shift towards a simplified convergence layer focusing directly on digital content access and distributed storing. ICN is considered both (i) as a *clean-slate approach*, running on top of the MAC layer, and (ii) as an *overlay approach*, running on top of the IP stack.

An interesting domain in which ICN is being studied as a clean-slate approach is the Internet of Things (IoT), e.g. in [1]. The IoT has already started being deployed, and will consist in large part in the interconnection of tens of billions of resource-constrained communicating devices [2], e.g. smart sensors and actuators of various kinds, a.k.a *Things*. This deployment is expected to generate massive amounts of data that will both (i) allow the optimization of existing processes, e.g. large-scale complex industrial processes, and (ii) support entirely new mechanisms and businesses based on the simultaneous availability of this data and ever increasing environment automation.

However, lessons learnt so far in the IoT point towards conflicting requirements concerning MAC layers. On the one hand, wireless communications are mandatory to provide the necessary cost-effectiveness and flexibility prohibited by the deployment and maintenance of too many wires. On the other hand, wireless communications are typically plagued with drastic reliability issues, compared to wired communications.

Time Slotted Channel Hopping (TSCH) [1] combines time division multiple access (TDMA) with frequency hopping. In contrast to CSMA, it thus implements a reservation-based medium access control. This may lead to improved performance, as long as the reservation schedule complies with the actual traffic demands. In previous work [3], we showed that TSCH benefits from the *predictable traffic pattern* of NDN. Our preliminary results indicated reduced energy consumption and higher packet delivery ratio compared to CSMA. In this paper, we discuss the problem and solution space as well as concrete design aspects for TSCH in ICN in detail.

The remainder of this paper is organized as follows. § II reviews related work in the domain of ICN, and states the problem we focus on: matching ICN paradigms and IoT link layer characteristics. § III provides an overview of the architecture we propose for optimized operation of ICN over a TSCH-based wireless link layer. § IV focuses on designing approaches for efficient scheduling of ICN Interest/Chunk traffic.

II. ICN and IoT: The State of Affairs

The IoT will include a large number of devices with constrained resources [4], which communicate via wireless channels. Deploying ICN in this context may not only facilitate some applications, but also simplify the protocol complexity and increase network efficiency [1], [5]. However, wireless transmission between IoT devices is typically built on contention-based MAC protocols (CSMA is the archetype), which are unreliable, prone to collisions and high packet loss. When combined with ICN approaches, such as NDN on which we focus in this paper, this unreliability leads to a number or problems, described below.

A. Problem Statement

Significant unreliability with wireless MAC protocols impedes the generation of coherent pending Interest paths, and amplifies the problem of state de-correlation [6] of NDN stateful reverse path forwarding (RPF).

Retransmissions and overload—as well as de-localized content in an unknown topology—can add high latencies to the information centric request-response pattern and lead to unpredictably high RTT fluctuations [6]. For NDN in fluctuating wireless environments, a node cannot reliably estimate when it will receive a content chunk in response to a pending Interest, or when it should retransmit this Interest. A forwarded Interest may have been lost or delayed somewhere in the network, or the Interest was just not satisfiable anywhere in the network. Consequently, it is hard to set a reasonable timeout for retransmitting the Interest, without any knowledge about transmission delays.

A further concern lies in energy consumption due to (avoidable) activity over radio. Aside from error recovery, receiver capacities are quickly drained by excessive broadcasts that occur from frequent reconfigurations, or unsophisticated routing practice. A particular problem in the wireless IoT domain thus lies in lightweight re-configurations and seamless route acquisitions. The latter poses a specific challenge, since the space of named routable entities is particularly large in the ICN world [5].

The typically unreliable and fluctuating nature of wireless communication in the IoT thus has a strong impact on the functionality of an ICN layer. This motivates the search for an alternative link layer technologies, which allow more appropriate cooperative use of the radio. A promising candidate is Time Slotted Channel Hopping (TSCH) [7], [8], which replaces CSMA with a reservation-based MAC protocol, combining TDMA with frequency hopping.

TSCH can drastically increase the reliability of packet transmission [9] thereby guaranteeing a fixed throughput and maximum latency even at high traffic load—if a proper schedule exists. However, an a priori derivation of a schedule requires thorough understanding of future traffic flows in the network which is infeasible for most application domains. Furthermore, traffic patterns and profiles may vary over time, leading to largely fluctuating demands that contradict the approach of a static schedule. In general, TSCH allows for a dynamic slot scheduling, but schedule negotiations are expensive. An approach for improving wireless ICN by TSCH thus poses the challenging problem of deriving and maintaining an adaptive scheduling of communication slots at an affordable cost.

B. Related Work

ICN has been identified as potential key enabler to improve reliability and security by design in wireless environments [5], [10]. For IoT scenarios, Li et al. [11] analyzed that ICN solutions which base forwarding on a global resolution service achieve comparable performance with ICN schemes based on reverse path forwarding (RPF), such as NDN. Baccelli et al. [1] showed that ICN can be implemented even on very constrained devices, and that ICN leads to performance gains compared to the currently standardized IoT protocol suite. To the best of our knowledge, however, there is no work on improving network conditions for the IoT by adapting the *MAC* layer based on principles of RPF-based ICN solutions.

Amadeo et al. [12] propose an NDN forwarding engine which allows for reliable multi-source data retrieval in IoT scenarios. They achieve collision avoidance on the *network* layer as consumers compute a random contention window for transmission. In this paper, we concentrate on reservation-based approaches on the *link* layer for the sake of robustness and efficiency. Furthermore, it is worth noting that our approach operates below the network layer, which leads to the following benefits. First, it abstracts from specific NDN implementations and thus broadens deployment. Second, it directly controls the duty cycling of the network controller. This is crucial with respect to energy saving because it enables to switch wireless cells off aligned with data transmission requirements. It also eliminates radio interference. TSCH multiplexes in time and frequency. Having control over the frequency per node is particularly important for IoT scenarios, where no infrastructure-based controlling of the wireless spectrum can be assumed.

Based on the observation that subsequent data chunks may vary significantly, Arianfar et al. [10] propose the assignment of an explicit lifetime to ICN packets to improve resource management at nodes as well as within the ICN network. The lifetime is derived from application requirements. Such information could be used to specify scheduled TDMA more precisely. However, in this paper we focus on a very basic adaption which does not require additional meta data. Furthermore, we follow the current implementations of CCN/NDN and thus consider the lifetime meta data as an optional optimization in future work.

III. The Idea of ICN over TSCH

A. The Potentials for Link-Layer Adaptation

(1) NDN Traffic Patterns Content distribution in NDN follows a request/response pattern with footprint on each hop. A request is propagated hop-by-hop in an Interest packet and implements a Pending Interest (PI) state in the corresponding tables (PITs) of intermediate nodes. Such a PIT entry matches at most one data chunk of limited size. Hence, in a fully deterministic, lossless setting, each request is answered by a train of up to k data packets within a time frame bound by the (temporal) diameter of the network.

For scheduling the wireless, we can interpret an Interest as a predictor of data expected on the reverse path, and conversely can exclude any data arrival in the absence of PI state. We can further exploit the predefined chunk size for fixing the ratio of data per Interest packet in our schedule. Ideally, the arrival of an Interest would trigger the allocation of k slot frames towards the appropriate neighbor at the expected time.

However, the dynamic reservation of cells requires coordination among neighbors and cannot be efficiently implemented chunk-wise. Neighbor selection furthermore assumes unambiguous routing information in place, which is often exceptional in IoT environments. We will show in the following sections how to procure routing and adapt scheduling in an efficient manner.

(2) NDN Faces NDN introduces the concept of faces as an abstraction of logical network interfaces between neighboring nodes. Faces map to point-to-point links in a typical wired environment. In low power wireless networks, though, nodes with omnidirectional antennas participate in shared links between a group of neighbors. Neighbor-specific faces (e.g., L2–tunnels) without isolation on links cannot freely co-operate, but will interfere with each other.

The use of a transmission schedule in TSCH allows to establish a cell-to-face mapping, while each cell (except for broadcast) is assigned to allow (unidirectional) transmission between individual nodes, only. Consequently, all scheduled cells within the transmission matrix of a node can be mapped to the corresponding faces. Each face (except for a broadcast face) will typically consist of at least two cells—one RX (receive) cell and one TX (transmit) cell.

Frequency division multiplexing in TSCH enables data transmission within multiple cells at the same time. Spreading channels among faces will allow to schedule several faces in parallel. A node can thus be enabled to communicate with several neighbors in the same timeslot.

B. Design Aspects and Requirements

In our following design, we focus on a typical IoT deployment scenario of a multi-hop wireless network that can reach the Internet via at least one gateway. While the nodes may be constrained, the gateway is assumed to have sufficient memory resources for holding a full FIB. Furthermore, we assume a fairly static topology with mostly stationary nodes, since mobility is not in the focus of IEEE 802.15.4e [13].

A use of ICN on TSCH in a network requires the following basic coordinative elements of TSCH in place.

Time Synchronization Operating the TDMA transmission schedule in TSCH requires a synchronisation of clocks within a low millisecond range. The maximum required precision is mostly derived from the *guard time*, which is, for example, set to 1.5 ms in OpenWSN, the de-facto reference implementation of IEEE 802.15.4e. Common IoT nodes with cheap oscillators exhibit a clock drift that can exceed $30\ ppm$, which poses high requirements on a clock synchronisation protocol. However, the required synchrony in time can be achieved either out-of-band (e.g., using a GPS signal), or by state-of-the-art clock synchronisation protocols for low power networks, such as the Gradient Clock Synchronization Protocol (GTSP) [14] or an adaptive synchronization towards a root node in tree-based topologies.

Frequency Coordination When calculating a schedule, each node needs to be aware of all neighbors that are in transmission range, so that unwanted overlaps in the time-frequency domain can be reliably avoided. For an efficient scheduling of nodes, a space-frequency division would be obstructive, and hence nodes need also knowledge about their two hop neighborhood. This information should be provided from topology building and used by a reservation protocol that is needed for negotiating the schedule among neighbors [15]. Both protocols can operate below the network layer and without interfering with NDN.

C. Topology and Routing

(1) Initializing a DODAG For successfully scheduling in frequency and time, we first need to create a topology within the network of IoT nodes. We propose to follow the common approach of building a tree-like structure—a destination-oriented directed acyclic graph (DODAG)—as known from RPL [16] with the IoT gateway in the role of the root node. Parents broadcast their presence (DIO) and children attach (DAO). These link-local operations can be transfered to the link-layer in a straight-forward manner. To facilitate frequency coordination, it can also be easily extended to inform about 2-hop neighbors.

Given this basic topology, every node can identify up- and downward paths and thus reach the gateway (root). We now need to address the more delicate question about arbitrary ICN routing on names. Here, we need to face the trade-off that Interests in a scheduled environment best float on a precise paths, but intermediate nodes have limited memory and cannot hold large routing tables.

(2) Learning Routes to Names In our previous work [1], we have designed and analysed two routing mechanisms—Vanilla Interest Flooding (VIF) and Reactive Optimistic Name-based Routing (RONR). While VIF works without a FIB, RONR nodes gradually acquire FIB entries in a reactive fashion. Given the DODAG topology, we will now follow the PANINI approach [17], [18]—an optimized strategy for routing Interests on names that makes a hybrid use of both routing primitives.

We select the gateway as the routing core under the previous assumption that it can hold a full routing table. Every node that offers a routable name advertises this name to the gateway. These Name Advertisement Messages (NAMs) travel hop-by-hop towards the root, and every intermediate node is free to update its own routing table. Intermediate nodes are not required to have a full FIB, but rather aim at adapting a few FIB entries to optimize guidance for Interests. Thus, each node autonomously decides about (a) its memory resources dedicated to the FIB, and (b) the forwarding logic it applies within its vicinity. Traffic flows can be continuously used to adapt the FIB to relevant traffic patterns. For example, a node can hold more specific information for frequently requested names, while it may erase entries for rather unknown traffic.

(3) A Bimodal FIB The objective of the FIB at intermediate nodes is to optimize traffic flows at minimal storage cost. For this, we propose to extend the FIB structure to hold two modes—`include` and `exclude`. In `include` mode, all Interests that match a FIB prefix will be forwarded on the associate Face, while all Interests that match a FIB `exclude`-prefix will be blocked on that Face. The initial state of an empty FIB reads `include *` which leads to a transparent forwarding (flooding) of all incoming Interests. A node that has seen no routable names from NAMs in a subtree of his may as well switch to `exclude *`. Based on this bimodal mechanism, typical optimizations could be as follows. An intermediate node sees much traffic of names with a prefix `/light/*` from many of its children, but some subtree(s) does not provide `/light/`-data. Assigning a single `exclude /light/*` to the corresponding Face(s) may result in an efficient trade-off between FIB memory and unwanted Interest traffic. A particularly effective optimisation can take place, if a node knows about `/light/*` in downward direction. It can place `exclude /light/*` at the upstream keeping all corresponding traffic local.

It is noteworthy that in the machine-to-machine oriented setting of the IoT it is easier to arrange names and topology in an aggregatable fashion, so that short prefixes may be effective for large collections of IoT data sources.

(4) Routing to Names After initial NAMs have arrived at the gateway and in the absence of any distributed routing knowledge, all nodes can reach all names by transmitting the Interest upwards. If an Interest cannot be satisfied on path, it will travel upwards to the root node, where it is flooded down its proper subtree. Even though suboptimal, this default routing is surprisingly lean [18]. Note that every node throughout the network can always tell whether an interest travels upwards, or downwards and thus can restrict flooding.

In the presence of meaningful, distributed FIBs, both routing phases benefit from optimization. Each hop on the upward path can redirect an Interest downwards to a local subtree, if a matching FIB entry exists. In the downward flooding phase, every request-related FIB entry narrows the dissemination of an Interest, and in the ideal case leads to a unique shortest path to the named data provider. It is worth recalling that nodes can adapt routing precision to traffic patterns so that frequently requested names or prefixes become more present in relevant FIBs.

IV. Scheduling

We now describe the design of a schedule for TSCH that is compliant to the ICN traffic pattern and adaptive to data demands. This shall flexibly optimize network performance and minimize energy consumption, but must not increase complexity for node coordination (see § III).

The general idea is a schedule that is partly static and pre-reserved, and partly dynamic and adaptive to the current traffic pattern. For this, we divide the slotframe into three parts, henceforth called subslotframes (*SSF*s). The first *SSF* is dedicated to statically scheduled Interest propagation and named SSF_I. Second, SSF_C is for sending back content chunks on a semi-dynamic schedule. The schedule of the third *SSF* is fully dynamic. This SSF_{Dyn} is activated to serve increased traffic loads on dedicated links.

For the following description of the scheduling procedure, we define $G = (V, E)$ as an undirected graph with a set of vertices V representing the set of nodes and a set of edges E representing the links between two nodes present in the routing graph. If two nodes a and b share an edge $(a, b) \in E$, they are called *1-hop neighbors*.

$\mathbf{SSF_I}$ – Static Interest Schedule The cells in this first subslotframe are reserved at network bootstrapping after the topology is created (or reconfigured). For reconfiguration purposes, the reservation of the first cell ($c(1, 1)$) is fixed to a general broadcast (of entire wireless range) and used to alert all nodes within wireless reach. Nodes that do not need to send any reconfiguration data, are required to switch to receiving mode for slot 1 at channel offset 1. Each node reserves a predefined number of TX cells to each of its 1-hop neighbors, and a matching RX cell (same slot number, same channel offset) for each TX cell a 1-hop neighbor has allocated towards it. In this way, basic capacities for exchanging Interests among neighbors are defined. The amount of reserved cells per neighbor can be chosen according to a priori knowledge of communication patterns—upstream (or default) routes may receive higher capacities, for example.

Additionally, a node should reserve cells for broadcasting to cope with incomplete routing information. Broadcast capacities may be aligned with predictable traffic patterns and available FIB memory. Interest broadcasts are limited to 1-hop neighbors and different from the general broadcast in cell $c(1, 1)$.

$\mathbf{SSF_C}$ – Semi-dynamic Content Schedule Each Interest is potentially answered by a content chunk. Taking this information into account and assuming a maximal chunk size of k packets, the content schedule in the second *SSF* shall be built as follows. For each RX cell in SSF_I, a node reserves k TX cells, and for each TX cell in SSF_I, a node reserves k RX cells. As such, the cell assignment does *not* require any negotiations between nodes, but is a direct consequence of the SSF_I, and static.

However, the nature of NDN traffic allows for an adaptive operation of the SSF_C. Initially, all reserved cells are deactivated, which means that the transceiver will not be switched on and the CPU may remain in energy saving mode. Node b activates k RX cells for a neighboring node a, after an Interest has been sent to a in SSF_I. These cells will get deactivated again, either after a content chunk was received from a, or when the PIT entry times out and is removed. By deactivating cells, energy can be saved from reducing idle listening and increasing the time the CPU can spend in sleep mode.

In the case of Interest broadcasting, these savings cannot apply. To limit broadcast reception periods, we assign shared cells to SSF_C. A TSCH shared cell operates CSMA/CA for increased flexibility at the price of reduced reliability.

$\mathbf{SSF_{Dyn}}$ – Dynamic On-Demand Schedule Cells in the third part of the slotframe stay unreserved at bootstrapping, and are only activated if traffic demands exceed the initially foreseen capacities. On a per link base, a balanced set of Interest and content cells are (de)allocated dynamically between two nodes and adapt the wireless spectrum to current utilization patterns. In detail, each node monitors the utilization of the (directional) links to each of its neighbors. Link utilization U is measured as the ratio between *used* cells c_u and *scheduled* cells c_s: $U = c_u/c_s$.

If the recent link utilization U_{cur} from node a to node b over a pre-defined time period T exceeds a predefined threshold U_{Th}, a and b reserve a preconfigured set of additional slots for sending/receiving Interests and content in SSF_{Dyn}. Thresholds and allocated slot sizes are parameters of the network that can be adjusted to meet deployment-specific criteria (see example below). Deallocation is performed after the U_{cur} falls below a certain threshold U_{Tl} in T. In this way, radio resources can be dynamically adapted to actual (bursty) traffic demands that may vary between node pairs, while low (regular) communication requirements allow for extended sleeping cycles in radio interfaces and thus enhance energy efficiency.

The dynamic adaptation of the schedule requires coordination between 1-hop and 2-hop neighbors. The information about a node schedule and the schedule of its 1-hop neighbors can be piggy-backed in ICN (Interest) traffic in a memory-efficient representation (such as bit fields). In this manner, a node will gain knowledge about the schedules of all nodes within its 1-hop and 2-hop neighborhood. This information serves as basis for reserving additional cells in SSF_{Dyn} by a link scheduling protocol like *LAMA*.

Example Assuming a typical building automation scenario nodes may request (period) configuration and software updates—e.g., provided by gateway acting as the root node (1) in the routing tree. Taking this knowledge into account, nodes will make more reservations in SSF_I for upstream packets. Let a slotframe consist of 101 slots (as proposed by the IETF 6TiSCH WG) and 16 channel offsets (according to the 16 channels available in IEEE 802.15.4). For simplicity we assume furthermore that $k = 1$. A sensible partitioning could be to assign 20 slots to SSF_I and SSF_C respectively. Depending on the network's density a node may reserve 1 (high density) to 9 (very low density) cells per neighbor in each of the first two SSFs. The remaining 60 slots—remember that the first slot is reserved for broadcasting—are assigned to SSF_{Dyn} and thus unreserved in the beginning. While the cells reserved in SSF_I and SSF_C may suffice the general requirements for fetching and delivering configuration information, it may happen from time to time that more data has to be delivered to the downstream nodes, e.g. in case of a firmware update. In this case, nodes will detect a high utilization of the cells in SSF_I and SSF_C and according make reservations for these links in SSF_{Dyn}. Hence, up to 30 additional cells may be reserved for Interests and content chunks respectively. After the firmware update is fully delivered to the affected nodes, reservations in SSF_{Dyn} can be deallocated again.

It can be seen that the sizes of ideally SSF_I and SSF_C should be kept considerably small and only ensure basic connectivity, in order to assign more cells to SSF_{Dyn}.

Preliminary Experimental Results We implemented our proposal for multiple MAC configurations in a proof-of-concept [3]. This included static, dynamic, as well as adaptive ICN reservation. Compared to ICN on common contention-based MAC layers, we observed that our approaches (i) offer similar delay and throughput performance, but (ii) are more reliable and more predictable, and (iii) can reduce energy consumption by almost 50%.

V. Conclusion

Prior work has shown the potential and demonstrated the feasibility of ICN in the IoT, running directly on the MAC layer. However, to fulfill this potential, the ICN network layer employed in this context needs to remain (i) lean and mean in terms of memory requirements and complexity, and (ii) allow advanced energy efficiency. In this view, combining ICN with a wireless MAC layer based on Time Slotted Channel Hopping (TSCH) is an enticing perspective. The advantage of such a combination is two-fold. On one hand, the reliability of TSCH avoids the need for complex error recovery mechanisms at the ICN layer. On the other hand, TSCH combined with appropriate dynamic scheduling allows to be more energy efficient, while not incurring more delays, compared to typical contention-based MAC layers. In this paper, we have thus designed mechanisms adjusting jointly, and on-the-fly, both TSCH time slot reservations and ICN Interest/Chunk multi-hop routing.

Acknowledgements This work was partly supported by a grant from the German Federal Ministry of Education and Research within the project I3.

References

[1] E. Baccelli, C. Mehlis, O. Hahm, T. C. Schmidt, and M. Wählisch, "Information Centric Networking in the IoT: Experiments with NDN in the Wild," in *Proc. of 1st ACM Conf. on ICN*. New York: ACM, 2014, pp. 77–86.

[2] A. Rayes, M. Morrow, and D. Lake, "Internet of things implications on icn," in *Collaboration Technologies and Systems (CTS), 2012 Int. Conference on*. IEEE, 2012, pp. 27–33.

[3] O. Hahm, C. Adjih, E. Baccelli, T. C. Schmidt, and M. Wählisch, "ICN over TSCH: Potentials for Link-Layer Adaptation in the IoT," in *Proc. of ACM ICN. Poster Session*. ACM, 2016.

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

[5] D. Kutscher *et al.*, "ICN Research Challenges," IETF, Internet-Draft – work in progress 01, 2015.

[6] M. Wählisch, T. C. Schmidt, and M. Vahlenkamp, "Backscatter from the Data Plane – Threats to Stability and Security in Information-Centric Network Infrastructure," *Computer Networks*, vol. 57, no. 16, pp. 3192–3206, 2013.

[7] I. P. W. Group, "Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)," IEEE, IEEE Standard for Local and metropolitan area networks IEEE Std 802.15.4-2011, 2011. [Online]. Available: http://standards.ieee.org/getieee802/download/802.15.4-2011.pdf

[8] I. P. W. Group, "Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs). Amendment 1: MAC sublayer," IEEE, IEEE Standard for Local and metropolitan area networks IEEE Std 802.15.4e-2012, 2012. [Online]. Available: http://standards.ieee.org/getieee802/download/802.15.4e-2012.pdf

[9] T. Watteyne, J. Weiss, L. Doherty, and J. Simon, "Industrial IEEE802.15.4e Networks: Performance and Trade-offs," in *IEEE Int. Conference on Communications (ICC), Internet of Things Symposium*. London, UK: IEEE, 2015.

[10] S. Arianfar, P. Sarolahti, and J. Ott, "Deadline-based Resource Management for Information-centric Networks," in *Proc. of the 3rd ACM SIGCOMM Workshop on ICN*. New York, NY, USA: ACM, 2013, pp. 49–54.

[11] S. Li, Y. Zhang, D. Raychaudhuri, and R. Ravindran, "A comparative study of MobilityFirst and NDN based ICN-IoT architectures," in *Proc. of QShine*. IEEE, 2014, pp. 158–163.

[12] M. Amadeo, C. Campolo, and A. Molinaro, "Multi-source Data Retrieval in IoT via Named Data Networking," in *Proc. of ACM ICN*. New York, NY, USA: ACM, 2014, pp. 67–76.

[13] M. R. Palattella *et al.*, "On optimal scheduling in duty-cycled industrial iot applications using ieee802. 15.4 e tsch," *Sensors Journal, IEEE*, vol. 13, no. 10, pp. 3655–3666, 2013.

[14] P. Sommer and R. Wattenhofer, "Gradient clock synchronization in wireless sensor networks," in *Proc. of the 2009 Int. Conference on Information Processing in Sensor Networks (IPSN '09)*. IEEE, 2009, pp. 37–48.

[15] Q. Wang, X. Vilajosana, and T. Watteyne, "6TiSCH Operation Sublayer (6top)," IETF, Internet-Draft – work in progress 00, 2013.

[16] T. Winter *et al.*, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," IETF, RFC 6550, 2012.

[17] T. C. Schmidt, S. Wölke, N. Berg, and M. Wählisch, "Partial Adaptive Name Information in ICN: PANINI Routing Limits FIB Table Sizes," in *2nd ACM Conference on Information-Centric Networking (ICN 2015), Poster Session*. New York: ACM, Oct. 2015, pp. 193–194.

[18] T. C. Schmidt, S. Wölke, N. Berg, and M. Wählisch, "Let's Collect Names: How PANINI Limits FIB Tables in Name Based Routing," in *Proc. of IFIP Networking*. Piscataway, NJ, USA: IEEE Press, 2016, pp. 458–466.