A Named Data Network Approach to Energy Efficiency in IoT

Oliver Hahm, Emmanuel Baccelli, Thomas C. Schmidt, Matthias Waehlisch, Cédric Adjih

A Named Data Network Approach
to Energy Efficiency in IoT

Oliver Hahm   Emmanuel Baccelli   Thomas C. Schmidt   Matthias Wählisch   Cédric Adjih
Inria   Inria   HAW   FU Berlin   Inria

Abstract—**In the IoT, the trade-off between content availability and energy efficiency plays a crucial role. In this paper, we propose an energy-saving approach leveraging distributed caching for IoT content, based on an information-centric networking paradigm. We extend the NDN protocol with a variety of caching and replacement strategies, and we discuss alternative approaches for extending NDN to accommodate such IoT use cases. Based on experiments on real IoT hardware in a network gathering hundreds of nodes, we demonstrate these caching strategies can bring 90% reduction in energy consumption while maintaining IoT content availability above 90%.**

I. Introduction

In the IoT, energy efficiency and memory efficiency play crucial roles. The standard approach to energy efficiency in the IoT consists in combining the techniques below:

- Energy efficient hardware with micro-controller and radio consuming energy in mW range and ultra-efficient sleep modes in nW range (energy harvesting techniques may also be applicable in some cases, but are not the focus of this paper).
- Low-power radio and MAC layers based on radio duty-cycling, aiming to reduce idle listening as much as possible, e.g. ContikiMAC [1].
- Network layer protocols that are less chatty e.g. 6loWPAN stack protocols adapting IPv6 to IoT [2].
- Content caching centralized in the cloud or on a proxy, e.g. CoAP / HTTP caching [3].

In particular, with this combined approach, there is no trade-off between content availability and energy efficiency because, while IoT devices sleep a large part of the time, content availability is preserved by a proxy (or the cloud) which caches the content. With this approach, sensors can fully benefit from radio duty-cycling mechanisms allowing less than 1% radio activity [4], which can achieve 100-fold less transceiver energy consumption in some scenarios.

However, in other IoT use cases, a designated gateway/proxy is unavailable most of the time. One example is *in-the-wild monitoring of plants, soils or animals*, which requires a large number of small IoT devices embarking sensors, disseminated in an area, e.g. on a meadow. Another example is *monitoring large storage location*, which requires a large number of scattered IoT devices with sensors tracking the state of monitored goods or machines. In such use cases, there is no designated proxy and the cloud is reachable infrequently, e.g. only when a drone with appropriate communication capabilities flies by, or

when an employee tours to check the area, carrying a tablet that polls sensors via radio communications. One might consider installing a designated gateway which can communicate locally with the IoT devices using a low-power radio interface, and with a high-power radio interface i.e. the uplink. However, prior work in the field of wireless sensor networks indicates that the gateway will typically also sleep a large part of the time to save energy and increase life-time. For example, in [5] authors show how wireless sensor networks can benefit from an 8-fold improvement in energy efficiency using such techniques combined with dynamic clustering.

In the above IoT use-cases, while radio duty cycling is still possible the standard approach combining sleeping and centralized caching is not applicable: a trade-off reappears between data availability and energy efficiency. In absence of a designated network element to centralize caching, an alternative approach consists in aiming for a dynamically distributed cache. The focus of this paper is thus to study mechanisms that dynamically distribute cached IoT content and allow IoT devices to be in sleep mode as often as possible, while maintaining acceptable levels of availability for IoT content.

A. Related Work

Early work in the field of wireless sensor networks proposed on-path caching [6], [7] to reduce the need for end-to-end retransmissions at the transport layer in multi-hop wireless sensor networks (WSN). In [8], authors proposed a content-aware diffusion mechanism for WSN leveraging on-path caching. A similar approach is recently the focus of a growing community: the information-centric networking paradigm (for example NDN [9]) proposes communication that is not host-centric and conversational such as with TCP/IP, but content-centric and completely connectionless. With NDN, a data consumer requests content via *Interest* packets forwarded in the network, which eventually hit either a cache containing a chunk of this content, or the content producer itself. Either way, content chunks are sent back to the consumer via reverse path forwarding, following the trail of transient Pending Interest Table (PIT) entries maintained in each intermediate node. Such nodes may then opportunistically cache such content chunks in their Content Store (CS). While initially proposed for a wired, core Internet context and large non-transient content, information-centric networking (ICN) is most recently being considered for the IoT and small

transient content. In [10], authors experimented with NDN on an IoT testbed and hinted at potential memory- and energy-efficiency gains with such an approach compared to the traditional 6LoWPAN approach, but stopped short of studying caching and replacement strategies. In [11], a new NDN communication pattern for IoT (and an optimisation) is proposed to exploit the wireless broadcast nature of IoT networks to retrieve content from multiple producers with a single interest. Persistent PIT entries are used. In [12] authors propose complementary mechanisms to adapt NDN to information freshness requirements specific to IoT sensor data, by introducing mechanisms for consumers to express them. In [13], authors give a high-level overview of advantages, trade-offs and challenges of information-centric networking for IoT. With the purpose of tracking content changes, authors in [14] propose ChronoSync, a mechanism built on top of NDN architecture to efficiently synchronize data and data updates among multiple users (e.g. for distributed chat, distributed file synchronization). The closest related work is [15] which includes a study of a basic random caching strategy with LRU and observe performance gains in content delivery, via simulations on a grid topology.

In general, NDN can be used over various transports including IP or TCP. However, the typical constraints of IoT systems in terms of memory constraints and link layer packet size limitations, demand a deployment directly on top of the link layer [10]. Hence, while other services on top of an IP-based IoT stack could offer similar mechanisms as the ones described in the paper, in order to reduce energy consumption while maintaining the content availability, a NDN approach implements these mechanisms directly as part of the network layer and is therefore much more applicable for highly constrained IoT scenarios.

To the best of our knowledge, there is no prior work on advanced distributed caching strategies in IoT on real hardware, that addresses the trade-off of data availability and energy efficiency. In this paper, we focus on this problem. The contributions of this paper are the following:

- We extend NDN with mechanisms allowing various caching and replacement strategies combined with deep sleep strategies,
- We comparatively evaluate several caching and replacement strategies based on extensive experiments on real IoT hardware in a network gathering hundreds of nodes
- We demonstrate experimentally on real IoT hardware that NDN enhanced with these sleeping, caching and replacement strategies can bring 90% reduction in energy consumption, while maintaining the availability of recent IoT content above 90%.
- We discuss the impact of specific IoT scenarios on NDN applicability, and alternatives for NDN extension catering for such scenarios.
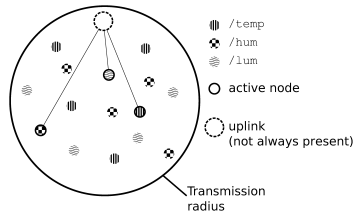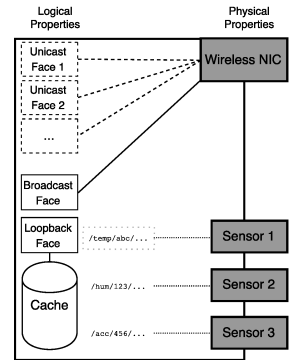

Figure 1. Broadcast domain.

Figure 2. IoT device.

II. IoT Scenario

We base the following discussions on a basic IoT scenario: a single wireless broadcast domain that gathers a set of sensors of various types as shown in Fig. 1. This domain is connected to the Internet via an intermittent uplink.

In detail, each sensor is a content producer, and is hosted on an IoT device as shown in Fig. 2. Each IoT device provides a small cache (RAM $\approx$ 50 kB) and a low-power CPU, to which are connected peripherals including a low-power radio interface. Such an IoT device is thereafter called a *node*, and a node can host one or more sensors.

Nodes alternate between *active* and *sleeping* phase according to a sleeping strategy, which can be either coordinated or uncoordinated. Upon generation of new content, a sensor can wake up the node it is hosted on (functionality available IoT operating systems, e.g. RIOT [16]), and can push this content to replicate it in the caches of other nodes that are currently in active phase. Nodes in active phase cache new content in their content store, the details depend on caching and replacement strategies (see Section III). When the uplink turns up, all active nodes transmit their cached data to the uplink. We focus on scenarios where sensors monitor a phenomenon whereby (i) data relevance strictly decreases with time, and (ii) a more complete view of what the sensors are monitoring is achieved if available data comes from a larger number of distinct sources (i.e. sensors).

In the subsequent sections, we discuss two crucial aspects in this context, the sleeping strategy and the cache replacement strategy.

III. Design Space: Strategies for Sleeping, Caching & Cache Replacement in NDN IoT

Our goal of deploying NDN in the IoT is to improve energy efficiency while maintaining availability of recent data. The core questions that need to be addressed are (i) how to organize sleeping of nodes to best use scarce energy resources, and (ii) how to organize cache maintenance as memory is limited?

We define and analyze a number of approaches for sleeping and caching. We distinguish between uncoordinated and coordinated approaches. In the **uncoordinated approach** nodes sleep in a randomized manner: each

node sleeps with probability p for a given period of time. Sleeping nodes rely on the fact that p is chosen in a way that the expectation for active nodes is ≥ 1 at any given point of time to receive and cache their data on their behalf. In the **coordinated approach**, a deterministic mechanism assigns only one node (per prefix and per period of time) which is then active and responsible for receiving and caching the data of sleeping nodes.

A. Uncoordinated Sleeping, Random Caching

In completely uncoordinated environments we cannot assume an administrative authority which pre-configures nodes. Each node thus decides every xD seconds whether it will be active or sleeping for the next xD seconds. The probability for sleeping mode is given by the parameter p. Active nodes that receive a content chunk will try to cache it with probability $q = 0.5$, similar to the caching approach depicted in [15]. The cache replacement strategy is a First-In First-Out (FIFO) policy, roughly equivalent to Least Recently Used (LRU) in this context.

B. Coordinated Sleeping

Single Deputy. This approach leverages coordination of nodes' sleeping phases. During the network's bootstrap, nodes determine an absolute order between them. The node in the first position of this order is elected as the first *deputy* and stays active for a certain period. Based on the determined order, each node will successively become *deputy* following a round-robin scheme. When a node wakes up to become the next deputy, it takes over deputy role by requesting the full cache from the previous deputy (using the simple Interest-based mechanism used by the uplink to request all available content from active nodes).

Multiple Deputies. In scenarios where the amount of relevant content exceeds a single cache, a single deputy is not sufficient. Hence, we introduce multiple deputies responsible for different prefixes like */hum*. This can be pre-configured by the network operator.

C. Name-based Cache Replacement

Max Diversity Most Recent (MDMR). To implement a cache that maximizes diversity of content wrt of different sensor sources, the IoT may benefit from the naming scheme in NDN. New content name is derived locally using the type of sensor (prefix) and the timestamp (suffix) see details in section V. The cache replacement strategy works then as follows: First, the cache tries to replace older chunks from the same producer. Next, the cache tries to replace the oldest chunk of a producer from which several chunks are present in the cache. Finally, if there is only entry per source, the oldest entry in the cache is replaced.

Prioritized Prefixes (P-MDMR). This prefix can be autoconfigured using local information coming from the (main) sensor of the node. E.g., a node with a temperature sensor will prioritize content for prefix */temp*. A node always tries to cache content chunks for the prioritized
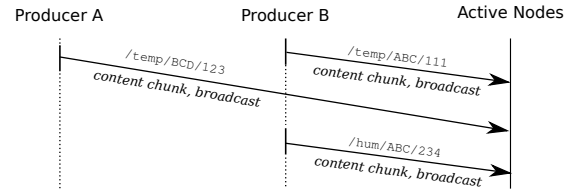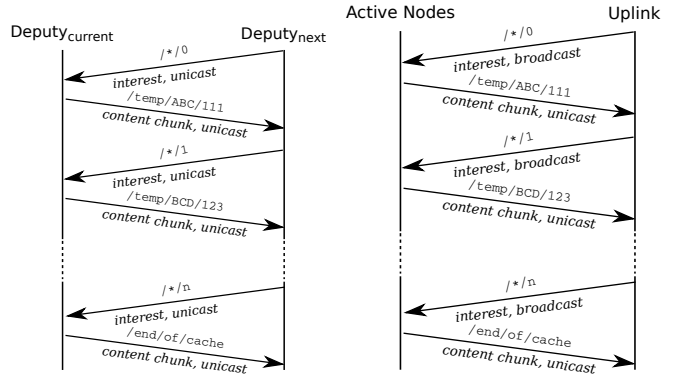

(a) Distributing data from sensors to cache as unsolicited content chunk.
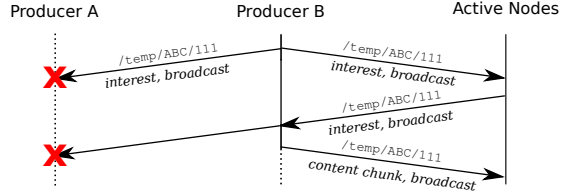
(b) Distributing data from sensors to cache using Interest-Interest.

Figure 3. Basic communication schemes between content producers and caches.

(a) Distributing data between two caches. (b) Distributing data from cache to uplink.

Figure 4. Basic communication schemes among deputies and between deputies and uplink.

prefix, while other content are cached with a probability $q < 1.0$. Fig. 1 depicts a network with three different types of sensors and consequently three different prioritized prefixes scattered in the local IoT network. If the cache is full, entries for non-prioritized prefixes are replaced first.

D. Basic Implementation Requirements

To implement the mechanisms described above in common IoT scenarios (see Section II), little changes are necessary based on NDN:

Opportunistic Caching of Unsolicited Content. When producers wake up, they want to offload content and immediately go back to sleep in order to save energy. The most basic approach is the distribution of new content via broadcast. In consequence, active nodes need to accept such broadcasts and allow for opportunistic caching of pushed, unsolicited content.

Several approaches for that are possible. For the experiments, we used both *Interest-Interest* [17] and an Immediate Broadcast (*IB*) approach similar to [18], which allows *producers* to immediately push their new content (active nodes have a permanent *PIT* entry matching the wildcard

prefix /∗ and hence, do not discard unsolicited content chunks). Note, that although *IB* does not comply with the "self-regulation of traffic" principle of NDN, it does not harm the flow-balance of the network, since it only occurs between two nodes and is never forwarded. We observed that, the main advantages of *Interest-Interest* compared to *IB* is that the Interest handshake conforms more easily with basic NDN, and increase robustness in face of packet loss typically experienced on IoT link layers, when this link layer is not too congested. The main drawback of *Interest-Interest* compared to *IB* is that it incurs significantly more control traffic and can thus be both (i) less energy efficient and (ii) counter-productive in terms of packet loss if the link layer is already congested. In particular, *Interest-Interest* cannot be used with *uncoord. sleeping* because (potentially) too many active nodes answer the initial Interest. Instead, preventive retransmissions (e.g. broadcasting 3 times upfront) can improve robustness against potential packet loss.

Interest Signaling for Group of Content. When a new deputy wakes up, or when an uplink appears, cached content should be transferred. However, nodes may not be aware of previously distributed content and therefore cannot request each content chunk explicitly. To request data for an unknown name, we require a wildcard symbol (e.g., `<prefix>/*`), which expresses interest for all content under the prefix.

Having those mechanisms in place, nodes can locate and transfer content without relying on strictly synchronized schemes or significant prior knowledge. We summarize the traffic exchanges between producers and active nodes in Fig. 3, and between active nodes and uplink in Fig. 4. The next deputy respectively the uplink request the whole CS from the current deputy chunk by chunk using the wild-card symbol. In the coord. sleeping approach a dedicated link layer address can be assigned to the current deputy, allowing for a unicast Interest. The uplink however must always use broadcast for its Interests.

IV. Experimental Evaluation

A. Content Availability Metrics

The goal of this paper is to study mechanisms that allow high sleeping/active ratio (p) while tending towards, or achieving atomic availability. Basically, this means that when the uplink requests content, recent content is available as if all nodes were in active state. We evaluate the availability of *recent* IoT content based on two metrics.
Diversity Metric:

$$Diversity := \frac{|C_{disj}|}{|S|} \qquad (1)$$

where $|S|$ is the number of content producers (i.e. sensors) in the network, and $|C_{disj}|$ is the number of disjoint name prefixes present in the aggregated cache C_{AGG} of all the currently active nodes.
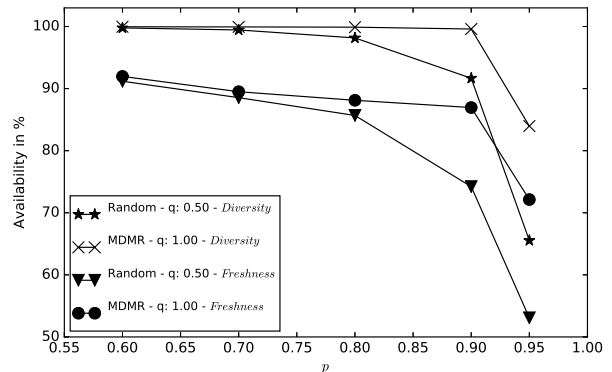

Figure 5. Availability for *uncoord. sleeping* approaches with different values for p wrt *Diversity* and *Freshness*. Experiments were conducted with 50 nodes.

Tolerated Freshness Metric:

$$Freshness := \frac{\sum_{i=1}^{|C_{disj}|} \mathbb{1}_{(c_i|AGE(c_i)<tD)}}{|S|} \qquad (2)$$

where c_i is an entry in C_{AGG}, $AGE(c_i)$ its age relatively to the newest entry from the same source, and t is a threshold value defining "tolerably" recent content. For the rest of the paper we will use the terms *Diversity* and *Freshness* to refer to the above defined metrics.

B. Implementation

Implementation with RIOT and CCN-lite. We implemented the caching, replacement, and sleeping strategies on top of RIOT [16] [1]. RIOT supports the NDN implementation of CCN-lite [19] as a *package*. We used some hooks in the CCN-lite protocol engine to implement small modifications to the processing of Interests and content chunks. For instance, a mechanism similar to the `mod_rewrite` module on HTTP daemons was introduced to rewrite Interests for / ∗ /N on active nodes to match the *Nth* entry in the content store.[2]

Experimental Setup on FIT IoT-LAB. The experiments were conducted on ≈ 220 nodes deployed over a 225 m^2, which are part of the Lille site of the FIT IoT-Lab testbed [20]. Each run lasted for 20 minutes, *D* was set to 3 *s* and *xD* to 12 *s*. Nodes were of the *M3* type, which are equipped with an 32-bit ARM Cortex-M3 MCU, 64 kB of RAM, 256 kB of ROM, an IEEE 802.15.4 2.4 GHz radio transceiver and four different sensors (light, accelerometer, gyroscope, pressure). In order to evaluate the energy consumption for these experiments, we measured the duration that a node spent in active and sleeping state plus the number of unicast and broadcast transmissions. Based on these hardware and network characteristics, we derive a simple energy consumption model: $E = \sum_{state} P_{state} \cdot t_{state}$, where P_{state} defines the power consumed for a given *state* and t_{state} is the time to spent in this state. We define the states: *sleeping*, *active* (listening

---

[1]Source Code is available at `http://ndnrg.riot-os.org/ccnl_caching`
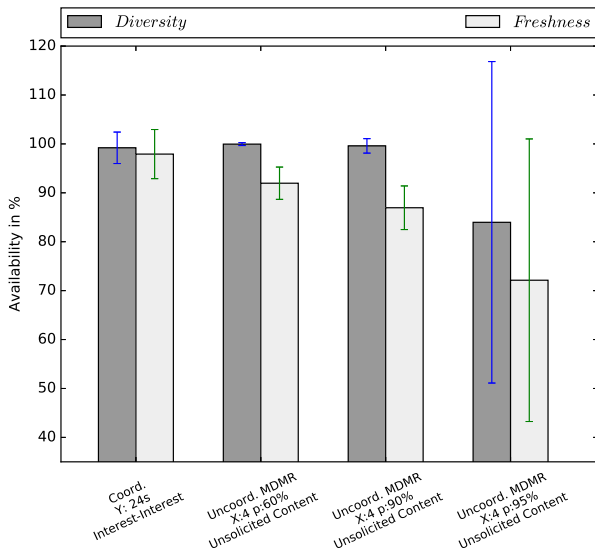[2]Chunks in the *CS* of each node have a arbitrary, but fixed order.

Figure 6. Comparing Availability wrt *Diversity* and *Freshness* between *coord.* and *uncoord. sleeping* approaches. Experiments were conducted with 50 nodes, Y represents the *deputy* cycle interval.

and receiving), *sending unicast*, and *sending broadcast* packets. Values for power consumption per state are taken from the datasheets of the MCU and radio transceiver. Furthermore, we assume a typical RDC rate of 0.6%, i.e. the default value for ContikiMAC.

C. Experiment Results

We evaluate two different cases: (i) the case where the cache of single node (size C) is bigger than the number of producers ($|S|$), and thus a single active node could cache all the recent content, and (ii) the case where C is smaller than $|S|$, i.e. multiple nodes must be active to cache all the recent content. In the second case, p must be chosen in a way, so that the expectation $(1 - p)|N| \cdot C \geq |S|$.

Small number of producers. First, we consider the case where $C \geq |S|$. We compare how *uncoord. sleeping* approaches perform for different values of p. We analyze only availability here, since an analysis of the energy consumption does not reveal any surprises: the time spent in active mode can be derived from p and the number of sent packets do not differ for the different approaches. It is worth noting that sensors hosted on currently active nodes also broadcast their newly generated data.

In Fig. 5 we notice surprising performance until a sharp drop in availability beyond $p = 0.9$. In detail, the *Random Caching* strategy achieves good diversity, with values just below 100% *Diversity* with $p \leq 0.9$, but drops below 70% above that threshold, e.g. for $p = 0.95$. Comparatively, the *Name-based Caching* strategy consistently achieves 100% for $p \leq 0.9$ and still a decent value of $\approx 85\%$ for $p = 0.95$.

Furthermore, regarding the *Freshness* metric (see Fig. 5), we notice that availability also drops for $p \geq 0.9$, below 80% for *Name-based Caching*, and much less, around 50%, for *Random Caching*.
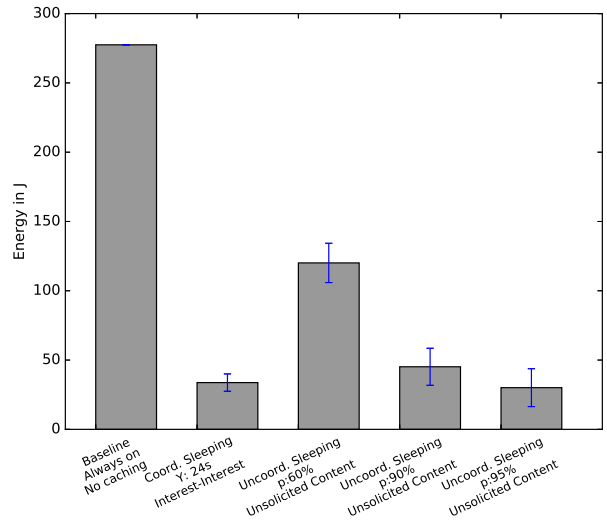

Figure 7. Comparing average energy consumption per node for *coord.* and *uncoord. sleeping* approaches.

Now let's compare these results with the performance of the *coord. sleeping*. Fig. 6 reveals that, compared to the *uncoord. sleeping* approach with $p \leq 0.9$, the *coord. sleeping* approach achieves similar diversity. However, in terms of *Freshness*, the *coord. sleeping* approach consistently outperforms *uncoord. sleeping* even when p is small.

In order to evaluate the energy consumption for these experiments, we measured the duration each node spends in active and sleeping state, and the number of unicast and broadcast transmissions. We then fed these values to the energy consumption model derived in Section IV-B. We compare energy consumption to a baseline where each node caches only the data it produces, has its CPU always on, but use state-of-the art radio duty-cycling with ContikiMAC. (Note that, hence, for the baseline, there is no communication between the nodes, only with the uplink). The results are shown in Fig. 7. With *coord. sleeping*, more network traffic is induced, but nodes can spend longer time in sleep mode, which compensates the energy consumption of the communication overhead. We observe that we can reduce the energy consumption by about 90% compared to the baseline without affecting the data availability (compare Fig. 6).

Large number of producers. Now, we analyze the content availability for the *uncoord. sleeping* approaches for scenarios where $|S| > C$. Fig. 8 presents these results for *Random Caching* and *Name-based Caching* strategies. We observe that availability (both *Diversity* and *Freshness*) decreases drastically with *Random Caching* and basic *Name-based Caching* strategies. To improve the availability, we use the P-variant leveraging *prioritized prefixes* to better use the capacity of the aggregated cache C_{AGG}. In detail, nodes will now cache content for an autoconfigured priority prefix with a probability of 1.0, and other content with probability 0.5. Using P-MDMR with 3 different types of sensors (i.e. 3 prioritized prefixes) availability is
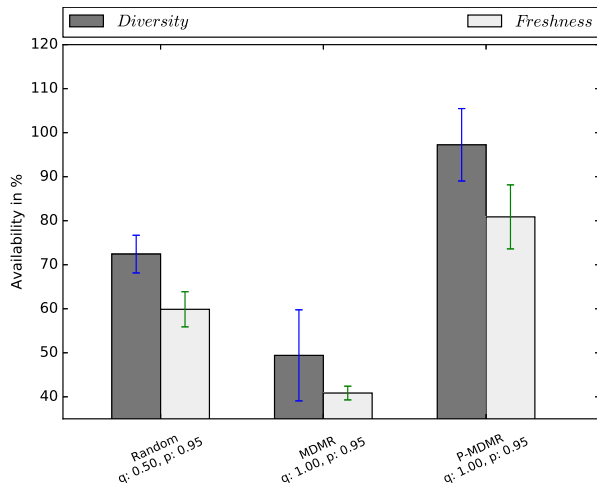
Figure 8. Availability wrt *Diversity* and *Freshness* for $|S| > |C|$ comparing different mechanisms for the *uncoord. sleeping* approaches. Experiments were conducted with 220 nodes and $p = 0.95$.

substantially better, both in terms of diversity (back above 90%), and in terms of freshness ($\approx 80\%$).

V. Discussion

Potential Energy Savings via Link Layer Name Mapping. Further energy savings may be achieved if *appropriate mapping of names to link layer* information could be leveraged. For example, one could map a specific name prefix to a specific PAN ID, and based on prefix priority, filter out radio packets in hardware, without having to wake up the CPU. This would be the equivalent of an overheard radio unicast being discarded at transceiver level, instead of at CPU level, which can thus remain in power-save mode and save further energy.

IoT Content Names. We considered that each sensor reading value fits in a single chunk and that each chunk can fit in one radio packet. Typical MTU on an IoT network is in the 100 bytes range (e.g. 127 bytes using IEEE 802.15.4). In practice, a sensor reading value coded on 32bits offers a quite reasonable range in most cases, and leaves substantial space for names. Therefore, while ICN schemes based on short names are necessary in this context, it does not seem a major blocker. For example, in our IoT deployment (as for any large IoT deployment), manual name configuration was out of the question. To bootstrap our deployment, names had to be autoconfigured. Each name must be derivable locally and must satisfy the requirements of (i) meaningfulness, and (ii) uniqueness. In order to satisfy the first requirement, we assumed a prefix that derived from sensor type identifier and a unique identifier of the node, e.g. a vendor ID. To fulfill the second requirement, we extend the prefix of the name by a suffix, the timestamp, which can also serve as a version number. The name could be enhanced by further information, e.g., based on geographical or organizational properties. A name generated by the autoconfiguration mechanism looks like $/hum/DEADBEEF/1466250645$.

VI. Conclusions & Future Work

In this paper, we have proposed and studied experimentally a number of mechanisms for name-based, decentralized, cooperative caching of IoT content, which allow to capture most of the phenomenons observed by IoT devices' sensors, while draining an order of magnitude less energy, compared to prior art. In practice, we adapted the NDN protocol for new IoT scenarios, and we extended NDN with novel caching and replacement strategies, which we have implemented and experimented on real hardware. Our testbed experiments show that these mechanisms achieve 90% reduction in energy consumption, while maintaining tolerably recent content availability above 90%. Our future work will focus on improving content availability even further while achieving similar energy consumption gains, when the number of nodes and the amount of new IoT content generated in the network grows larger.

References

[1] A. Dunkels, "The contikimac radio duty cycling protocol," 2011.
[2] Z. Sheng *et al.*, "A Survey on the IETF Protocol Suite for the Internet of Things: Standards, Challenges, and Opportunities," *Wireless Communications, IEEE*, vol. 20, no. 6, pp. 91–98, 2013.
[3] Z. Shelby *et al.*, "The constrained application protocol (CoAP)," Tech. Rep., 2014.
[4] D. Stanislowski *et al.*, "Adaptive synchronization in IEEE802.15.4e networks," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 795–802, 2014.
[5] W. R. Heinzelman *et al.*, "Energy-efficient communication protocol for wireless microsensor networks," in *System sciences, 2000. Proceedings of the 33rd annual Hawaii international conference on.* IEEE, 2000, pp. 10–pp.
[6] F. Stann and J. Heidemann, "Rmst: Reliable data transport in sensor networks," in *Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE. 2003 IEEE International Workshop on.* IEEE, 2003, pp. 102–112.
[7] A. Dunkels *et al.*, "Making tcp/ip viable for wireless sensor networks," *SICS Research Report*, 2003.
[8] C. Intanagonwiwat *et al.*, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," in *Proceedings of the 6th annual international conference on Mobile computing and networking.* ACM, 2000, pp. 56–67.
[9] V. Jacobson *et al.*, "Networking named content," in *Proceedings of the 5th international conference on Emerging networking experiments and technologies.* ACM, 2009, pp. 1–12.
[10] E. Baccelli *et al.*, "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.
[11] M. Amadeo *et al.*, "Multi-source data retrieval in IoT via named data networking," in *Proceedings of the 1st international conference on Information-centric networking.* ACM, 2014, pp. 67–76.
[12] J. Quevedo *et al.*, "Consumer-driven information freshness approach for content centric networking," in *Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on.* IEEE, 2014, pp. 482–487.
[13] A. Lindgren *et al.*, "Design choices for the iot in information-centric networks," in *2016 13th IEEE Annual Consumer Communications & Networking Conference (CCNC).* IEEE, 2016, pp. 882–888.
[14] Z. Zhu and A. Afanasyev, "Let's ChronoSync: Decentralized Dataset State Synchronization in Named Data Networking," in *2013 21st IEEE International Conference on Network Protocols (ICNP).* IEEE, 2013, pp. 1–10.
[15] M. A. M. Hail *et al.*, "On the Performance of Caching and Forwarding in Information-Centric Networking for the IoT," in *Wired/Wireless Internet Communications.* Springer, 2015, pp. 313–326.

[16] E. Baccelli *et al.*, "Riot os: Towards an os for the internet of things," in *Computer Communications Workshops (INFOCOM), 2013 IEEE Conference on*. IEEE, 2013, pp. 79–80.

[17] J. Burke *et al.*, "Securing instrumented environments over content-centric networking: the case of lighting control and ndn," in *Computer Communications Workshops (INFOCOM WKSHPS), 2013 IEEE Conference on*, April 2013, pp. 394–398.

[18] M. Amadeo *et al.*, "Internet of things via named data networking: The support of push traffic," in *Network of the Future (NOF), 2014 International Conference and Workshop on the*, Dec 2014, pp. 1–5.

[19] "CCN Lite: Lightweight implementation of the Content Centric Networking protocol," 2014. [Online]. Available: http://ccn-lite.net

[20] C. Adjih *et al.*, "FIT IoT-LAB: A large scale open experimental IoT testbed," in *Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on*. IEEE, 2015, pp. 459–464.