A down-to-earth integration of Named Data Networking in the real-world IoT

Amar Abane, Mehammed Daoui, Paul Mühlethaler, Hossam Afifi

A down-to-earth integration of Named Data Networking in the real-world IoT

Amar ABANE*†, Mehammed DAOUI*, Paul MUHLETHALER‡, and Hossam AFIFI§
*LARI Lab, University Mouloud Mammeri, Tizi-Ouzou, Algeria
Email: mdaouidz@yahoo.fr
†CEDRIC Lab, Conservatoire National des Arts et Metiers, Paris, France
Email: amar.abane.auditeur@lecnam.net
‡Inria, Paris, France
Email: Paul.Muhlethaler@inria.fr
§Telecom-SudParis, France
Email: hossam.afifi@telecom-sudparis.eu

***Abstract*—The IEEE802.15.4 wireless technology is one of the enablers of the Internet of Things. It allows constrained devices to communicate with a satisfactory data rate, payload size and distance range, all with reduced energy consumption. To provide IoT devices with a global Internet identity, 6LoWPAN defines the IPv6 adaptation to communicate over IEEE802.15.4. However, this integration still needs additional protocols to support other IoT requirements, which makes the IP stack in IoT devices more complex and therefore shows the limitations of the IP model to support the needs of future Internet. Named Data Networking represents an alternative that can natively support IoT constraints including mobility, security and human readable data names. This paper is a synthesis of an ongoing work that investigates the integration of NDN with IEEE802.15.4 for constrained IoT devices. The proposed design has been implemented in a real-world smart agriculture scenario, and evaluated by simulation focusing on energy consumption and network overhead in comparison to IP-based protocols.**

***Index Terms*—Named Data Networking, wireless networks, Internet of Things, IEEE802.15.4, Information Centric Networking, Precision agriculture.**

I. Introduction

The deployment of low power wireless technologies on affordable SOMs and single board computers have fostered the emergence of billions of devices, through which people and "Things" are becoming connected over the Internet. This new trend of cyber-physical Internet is commonly designated by *Internet of Things* (IoT [1]).

IoT systems are built with battery powered devices, which are computationally limited, mobile and massively deployed. Device interconnection is achieved with low-rate wireless technologies, which allow communication with a satisfactory data rate, payload size and distance range, all with years of battery lifetime. These wireless technologies provide the best compromise for resource-constrained devices. The IEEE 802.15.4 is the standard on which many low-rate wireless devices are based and its wide adoption proves its suitability for the IoT. Therefore, IPv6 integration with IEEE 802.15.4, known as 6LoWPAN [2], is designed to provide IoT devices with an IP layer-3 "identity".

Nevertheless, IPv6 integration needs additional protocols to support the IoT requirements in terms of security and mobility. The current complexity of the IP stack in IoT devices exhibits its limitation to support emerging IoT applications [3].

While efforts like 6LoWPAN try to make the IP architecture compliant with the IoT vision, the Information Centric Networking (ICN) approach natively supports the needs of the current Internet, particularly the IoT.

One of the embodiments of ICN is Named Data Networking [4], [5]. Instead of using source and destination addresses to deliver packets, NDN communications are completely based on content names. In NDN, the pieces of data are named independently of their location and requested directly from the network. Intrinsic features come along with this principle such as: communication without establishing end-to-end connections, without name resolution, native multipath routing and in-network caching. Consequently, NDN can address the IoT requirements at the network level and natively supports application semantics, data-centric security and (consumer-side) mobility.

So far, the suitability of NDN for the IoT has been investigated to some extent (Section IV), but its lightweight aspect remains to be improved in order to enable NDN in IoT applications. We observe that an NDN stack implementation in IoT devices (RIOT and Contiki) can save up to *60%* of ROM and *80%* of RAM compared to the RPL/6LoWPAN stack [6]. Moreover, a simple flooding mechanism in an NDN wireless network generates three times fewer packets than RPL routing with 6LoWPAN.

This paper draws a global picture of a real-world NDN integration in the IoT. We aim to show that providing IoT devices with a layer-3 data-centric identity can be more relevant than the current IPv6 integration in terms of overhead and energy consumption, while providing satisfactory performance. Therefore, some pragmatical NDN-specific operations are proposed to enable NDN in constrained devices over IEEE802.15.4. The NDN-802.15.4 architecture is deployed in a precision agriculture system.

In the rest of this paper, Section II presents the evolution of

the Internet that has led to NDN, Section III gives some details about the NDN architecture and Section IV reports on related work that investigate NDN in the IoT. Section V describes the NDN-802.15.4 architecture. Different aspects of evaluation are reported in Section VI and, Section VII concludes the paper with some perspectives.

II. From IP to NDN: a paradigm shift

Originally, the IP protocol was designed to provide a communication mechanism between two end-points, identified by numerical addresses. This host-based model was an extension of the telephone communication model to support data exchange. Therefore, IP focuses on deliver packets between a source host and a destination host. Above this model, additional layers were developed to support flow control, end-to-end reliability and user applications creating the Internet protocol stack. In the meantime, revolutionary Internet applications (e.g. www) shifted the focus from identifying hosts (IP) to identifying resources (URI) leading to the creation of the DNS system, CDN and so on. Consequently, two different namespaces are currently involved in the Internet protocol stack: IP addresses and the natural content names.

However, the emergence of the IoT puts once again the IP model to the test, and highlights its limitations to provide the "true" IoT functioning. Despite middleware and additional layers, security is focused on communication channels when the data itself needs to be secured. Most IoT devices rely on the Cloud to perform access control and security verifications while they should communicate independently and directly over the network. Moreover, IoT users are expected to request and retrieve data from anywhere, over all available network interfaces while keeping a unique identity and an optimal level of security.

In this context, the Information Centric Networking paradigm creates an alternative to support future Internet features, that has been highlighted by the IoT. ICN architectures consider the data as the first-class network entity.

Although it is an L3 protocol, NDN can run as an overlay of IP, UDP or TCP protocols. This provides an opportunity for using NDN in the current Internet infrastructure. Considering firstly such a deployment is more realistic, and provides a way to experiment NDN in a real context. However, multiple approaches can be envisioned to deploy NDN on the current network infrastructure. Studying these approaches is out the scope of this paper, we consider integrating native NDN in the edge of an IoT architecture (i.e. gateway-device) and NDN over UDP/IP in the backbone.

III. NDN overview

NDN uses two types of packets: an Interest to request content, and a Data that satisfies one Interest with a piece of content. Each NDN node needs three data structures to process packets: FIB (Forwarding Information Base), PIT (Pending Interest Table) and CS (Content Store).

Consumer-driven communication. An NDN exchange operates according to the following steps: (1) The consumer requests a certain content by issuing an Interest carrying the name of the requested data (e.g. */room1/temperature*). (2) The router that receives the Interest checks if the corresponding Data exists in the CS; if the Data is found it is returned to satisfy the Interest without going further. Otherwise, the router verifies if an Interest requesting the same content is already in the PIT: if so the Interest is dropped and only the incoming interface is added to the existing PIT entry (request aggregation). If there is no Interest waiting in the PIT, the router consults the FIB to find one or multiple interfaces to forward the Interest and creates a PIT entry containing the forwarded Interest with its incoming interface. (3) When the Interest reaches the node that contains the requested content, this node sends back a Data packet carrying the content, the name and security information about the content producer (e.g. signature, key name, etc.). This Data packet takes the reverse path of the Interest following traces left in the PIT of each intermediate node. Typically, when a relay node forwards back a Data packet, it saves a copy in its CS to reuse it to satisfy future Interests (caching).

Flexible packet format. NDN packets are encoded in the TLV (Type-Length-Value) format. Although Interest and Data packets have default and optional fields respectively, they do not have predefined packet size or field sizes. TLV encoding represents an NDN packet as a collection of sub-TLVs, without a packet header or protocol version. Each TLV block is identified as a sequence of bytes starting with a predefined marker, followed by its length and its value.

Natural naming. NDN contents are identified through URL-like hierarchical names. A name is formed by a sequence of components separated by slashes ("/"). For example, the name */home/room1/temperature* may identify the temperature value of room 1, while */home/room1/humidity* identifies the humidity value in room 1, and */home/room2/all* identifies all sensor values of room 2. Applications are free to design their own naming scheme since the routers do not interpret the whole name. Moreover, a producer can add name components to the initial name in order to provide more information about the content (e.g. timestamp, sequence number, geolocation, etc.).

IV. Related work

A. NDN meets the IoT world

Different IoT aspects have been investigated from the NDN perspective. In this section, we focus on propositions that provide real-world deployments.

The NDNoT project [7] is an NDN-based platform for home automation, providing features like: service management, control access, and data collection. The hierarchical structure of NDN names is efficiently examined and signed Interests are proposed to act as device commands. NDN-BMS [8] is a building monitoring system deployed at UCLA, that manages data publishing and consuming through an access control mechanism based on the NDN data-centric security. An

expressive human-friendly naming scheme has been adopted to provide a natural way to retrieve data. However, device/gateway communications use proprietary protocols while only data publishing and access control is performed with NDN. Authors in [9] and [10] propose a secured sensing framework for NDN-based sensors. Many sensing modes are envisaged such as push, pull and on-demand. The framework mainly addresses security issues through a trust model, a pairing protocol and an access control mechanism.

In the field of wireless sensor networks (WSNs), [11] deals with the use of Content-Centric Networking (CCN) for WSNs. Therefore, a lightweight version of the CCN protocol has been designed to accommodate constrained device requirements and small IEEE 802.15.4 frames. In the proposed protocol, restrictions are imposed on the length of packet fields and names.

Recently, the NDN protocol stack has been ported to the RIOT platform [12]. The implementation provides a basic support of the IEEE 802.15.4 technology with a simple fragmentation mechanism.

In a previous contribution, we proposed a one-hop NDN communication over ZigBee [13] in which the sensing features of ZigBee modules were integrated in NDN. The NDN-802.15.4 integration presented in this paper is more complete and it targets IEEE 802.15.4 with dedicated operations.

Other NDN contributions for the IoT are discussed further in [14].

B. Wireless forwarding with NDN

Multiple studies investigated NDN routing in wireless networks, mostly focusing on MANETs. The propositions summarized below can be seen as *"flood-and-learn"* mechanisms [15]; in the sense that an Interest flooding phase is used to discover content sources, and subsequent requests are forwarded more accurately based on the *learned* information.

In [16], a broadcasting protocol has been designed for data dissemination in vehicular networks, in which timers were used to defer packet forwarding. During the waiting time, if a node overhears a packet with the same name, the scheduled packet is dropped, which minimizes the collisions. However, this blind-flooding mechanism causes a large overhead and does not guarantee that the best path is used to retrieve content.

More elaborate solutions have been proposed to move from a blind-flooding to a controlled-flooding. In the *Listen First Broadcast Later* strategy (LFBL [17]), each node deduces the delay time before retransmission from its eligibility to forward the Interest. A node's eligibility is based on its distance from the content source that each node maintains: the closer to the source, the shorter the waiting time. This framework however, uses a distance table and endpoint identification, which slightly evokes the host-based communication model from which NDN differs.

In [18], defer timers are replaced by a forwarding rate adjustment. Each node collects its neighborhood forwarding statistics and periodically adjusts its forwarding rate (i.e. the ratio of the packets it should forward). This mechanism makes the nodes collaboratively forward packets without superfluous traffic and no additional data structure is needed. As it is based on a purely statistical method, nodes need to first get enough information and reactively adapt to changes in the situation.

In [6], a realistic improvement of the blind-flooding mechanism has been implemented in a network of IoT devices, with the aim of reducing network overhead and minimum routing state. After retrieving the first Data by Interest flooding, nodes keep a temporal FIB entry in order to avoid flooding subsequent Interests. If the FIB entry for the requested data does not exist, or it was deleted, the flooding phase is performed again to discover another content source. This simple modification significantly reduces the network overhead, but improvements can still be made.

V. NDN-802.15.4 INTEGRATION

The integration of NDN with IEEE 802.15.4 aims to make wireless devices an integral part of the NDN network. Therefore, a version of the NDN module should be considered for constrained devices, while providing them with all NDN functionalities.

Figure 1 depicts a typical IoT architecture considered for the NDN-802.15.4 integration. Each Low-rate Wireless Network (LWN) is accessible with a common prefix (CP). The architecture components are described below.

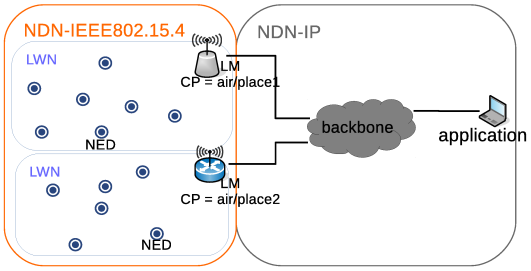

Fig. 1: IoT architecture for NDN-802.15.4 integration

A. Strategy layer

The strategy layer (Figure 2) implements procedures and concepts used to handle NDN communication over IEEE 802.15.4. The strategy layer operations are presented in the following:

Name-Payload-Field balancing. So far, propositions made to support NDN in constrained environments have been based on predefined restrictions. However, excluding some fields and limiting name length arbitrarily or intuitively is not suitable to cover all applications. For example, sensors that need to name small-sized data according to the location, the timestamp, the data type and the version, require maximizing the name length at the expense of the payload. Whereas, a livestock monitoring system that needs to identify each animal individually with all its related information will need to have data named with an ID and timestamp, but a larger payload is required for the data.

To support small-sized frames, it is important to understand the proportion in length between name, payload and fields in

a Data packet before generating it. Therefore, we propose a Name-Payload-Field function for constrained frame length.

The following notation is adopted to describe the function:

- F: payload length of the frame considered. (e.g. $F = 100$ Bytes)
- d: length of the Data packet with defined fields (except *Name* and *Content* fields) and all necessary Type-Length bytes
- p: payload length of the Data packet. ($p \geq 0$ Bytes)
- s: length of the signature value in the Data packet
- $f(p)$: size of the *Name* field according to the variable payload ($f(p) \geq 0$)

Here, a Payload-Name relation can be intuitively evaluated as:

$$f(p) = F - (p + d + s) \tag{1}$$

To find a relation between the structure of the Data packet and its size d, we study the possible field combinations and their corresponding sizes. For each possible field combination, we calculate the Data packet size including all the necessary bytes, except *Name* and *Content* fields. After calculating the Pearson correlation coefficient for the combinations obtained, the Payload-Name relation defined in Equation (1) can be improved to Payload-Name-Field relation, expressed as follows:

$$f(p) = F - (p + (8.93 + 4.36m) + s) \tag{2}$$

where, m is the number of fields that are defined in the Data packet considered.

The $f(p)$ function gives the expected size of the *Name* field according to the payload length and the number of optional fields. Hence, it can be used as a dynamic guider for applications and devices to set the appropriate naming scheme for data, while balancing it with payload size and packet fields.

Asymmetric packet compression. Interest packets from the backbone can rapidly become large, due to long names and optional fields. However, we observed that corresponding data from the LWN can be retrieved without sending all Interest fields. Missing fields can be calculated based on predefined configuration, NDN packet specification and previously shared information. Following the same principle, Data packets from devices can be sent with missing fields that can be generated by the gateway. Given this, we designed a simple packet compression scheme to reduce wireless network overhead.

We adopt a packet fields classification (Table I) according to which the packets are compressed, depending on whether devices maintain more or less contextual information with the gateway.

We observe that the proposed packet compression does not always require a decompression. Indeed, the LM and NEDs perform different (de)compression operations. When a NED receives a compressed Interest, it does not need to calculate all the missing fields to generate the data or to forward the Interest. Furthermore, if a decompression is required, each field can be extracted separately. When the response Data packet reaches the LM, it decompresses it and forwards it to the backbone. This packet decoding feature combined with asymmetric (de)compression help to save memory and reduce packet replications.

From an implementation point of view, the compression process consists in omitting/updating certain bytes before the packet is sent. The sender does not need to maintain the two states of the packet (i.e. compressed and decompressed). Similarly, the receiver decompresses the packet by adding/updating certain bytes.

Regarding security, data authenticity is not compromised by packet compression. When each NED signs its data, the initial (i.e decompressed) Data packet is signed before its compression, and the decompression process by the LM adds the exact bytes needed to make the signature verification correct. When the Data signature is delegated to the LM, the Data packet is signed by the LM after the decompression.

Fragmentation. In NDN, routers need the entire Interest and Data packets to perform NDN forwarding operations. Therefore, a hop-by-hop fragmentation is the only one possible when a packet is larger than the link MTU.

Alternately, applications can use NDN segmenting and/or sequencing to transmit and retrieve large data amounts separately, and without causing extra computation or additional header in the strategy layer.

However, Data signature is mandatory in NDN and it is frequently achieved with public key encryption, making it difficult to fit a Data packet with a payload and a signature into one IEEE 802.15.4 frame. With packet compression, we expect that fragmentation will be required only when public key cryptography is used. In this case, we adopt the 3-byte fragmentation header is proposed in [12].

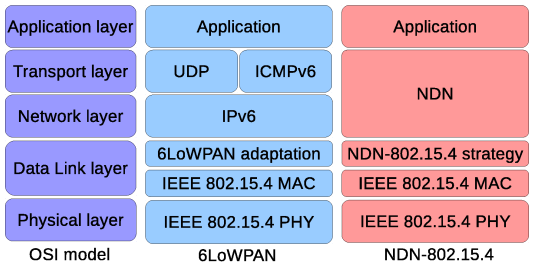

Fig. 2: NDN-802.15.4 stack, OSI model and 6LoWPAN

B. *Lightweight forwarding strategy*

By side of reducing packet size, a forwarding strategy is needed over the wireless network. Since routing operations are based on names in NDN, using it as an overlay on top of IP requires name resolution, which makes a lightweight implementation impossible. For this reason, we use NDN directly on the MAC layer in the wireless network.

The proposed forwarding strategy (denoted iLNFS) does not use device addresses. Instead, it follows the aforesaid flood-and-learn approach, and relies on broadcasts (and overhearing) to retrieve content based on names and time-deferred

TABLE I: Packet fields classification.

Class	Description	Fields	Treatment
STATIC	Shared by NEDs and the LM in the local wireless network	(part of) *Name*, *NameComponent*	Not transmitted
INFERRED	Not exactly the same for the LM and NEDs, but can be calculated using shared context and trust conventions	*SignatureType*, *KeyLocator*	Not transmitted
DEFAULT_VALUE	Fields with a default value defined in the NDN specification, or fields that can be omitted	*ContentType*, *FreshnessPeriod*, *HopLimit* *InterestLifetime*, *MustBeFresh*	Not transmitted when default value
VARIABLE	Can not be inferred and not common to LWN nodes	*Nonce*, *Content*, *SignatureValue*, *Parameters*	Always transmitted
UNSUPPORTED	Not supported because their processing is too complex for constrained devices	Rest of the fields (e.g. *ForwardingHint*).	Not transmitted

retransmissions. Unlike related solutions, the flooding phase in our proposal is triggered only once; when a new prefix appears in the network. As a consequence, the number of packet retransmission is reduced to the bare minimum without requiring to maintain additional information.

iLNFS is based on an adaptation of the Q-learning framework which is not described here due to space limitation. However, to reduce the network overhead and keep a minimal state in NEDs, random exploration and data structure used in Q-learning are replaced by a combination of partial-learning and real-time delay refining.

Considering the Q-learning framework, each node maintains a Q-value of the corresponding content prefix and updates it after retrieving (or overhearing) a Data packet, according to the following equation:

$$Q_x(p) = (1-\alpha)Q_x(p) + \alpha\left(r + \min Q_y(p)\right) \qquad (3)$$

where α is the learning rate, r is the reward [1], $Q_x(p)$ is the Q-value of node x for the prefix p, and $\min Q_y(p)$ is the smallest Q-value recently heard by node x.

After the first discovery of the content source with an Interest flooding phase, the effective strategy starts as follows: (i) the source node responds with a Data packet carrying the initial Q-value, (ii) the first forwarder on the source-consumer path computes its Q-value, replaces the Q-value with the value obtained, and retransmits back the Data packet, (iii) each node on the path follows the same procedure until the Data packet reaches the consumer, (iv) in the vicinity of the path, nodes that overhear the Data packet can perform a passive Q-value update to learn their position relative to the data source.

Subsequent Interest forwarding is then performed by combining partial-learning and real-time delay-adjustment phases.

The forwarding decision in a relay node consists in finding the appropriate delay to wait before retransmitting the Interest.

Let $Q_x(p)$ and $Q_y(p)$ be the current Q-values for the prefix p at nodes x and y respectively. Let $\Phi(a)$ be the delay to wait before retransmission, and $a = \Delta + \theta$. The definition of Φ will be given later.

Whenever node x receives an Interest issued (or forwarded) by node y, it computes the value $\Delta = Q_y(p) - Q_x(p)$. Here, Δ quantifies the global eligibility of node x to forward the Interest. If $\Delta \geq 0$ then node x can potentially forward the Interest and can evaluate θ to obtain the delay time before Interest retransmission.

θ is a statistical value used to adjust the time to wait according to the neighborhood activity.

Let N_a be the neighborhood activity rate for all data names. From the perspective of a node, N_a can be computed by $N_a = D_u/I_d$, where D_u is the number of unsolicited received Data and I_d is the number of non-forwarded Interests (dropped Interests). Then, θ may be simply defined as $\theta = Th - N_a$, where $Th \leq 1$ is the activity threshold above which the waiting time should be increased. For simplicity, but without losing accuracy, N_a is kept between 0 and 1. Thus, if Th is strictly lower than 1 (e.g. *75%*), θ can be negative. In this case, the value of Δ is reduced, which will increase the waiting time. When no statistic is available, $N_a = Th$.

After obtaining the appropriate value of *a*, the delay time is computed with the following function:

$$\Phi(a) = \frac{M}{e^{a/2}} + m \qquad (4)$$

Here, the waiting time is upper-bounded by (*M+m*) units when $a = 0$; *m* forces the forwarder to wait for a minimum period to prioritize a response Data packet if any.

During the delay-listening time, if node x detects that a forwarder z is transmitting a packet with the same name, it deduces that z is more eligible to handle the Interest and cancels its retransmission.

In order to deal with obsolete distance estimation, each node uses Interest timeout to reset the corresponding Q-value to a maximum value. After that, it will be able to update it with an accurate estimation.

VI. Evaluation

We split the evaluation into three parts: (i) an implementation measurement achieved in a real-world IoT deployment, (ii) a local-scale evaluation of the wireless forwarding strategy, and (iii) a full-scale overhead estimation in comparison to IPv6.

A. IoT deployment

The NDN-802.15.4 architecture has been deployed in a cow health monitoring system for precision agriculture. Specifically, the system uses sensors (e.g. movement, temperature, etc.) to collect the data of each cow. The data is then analyzed to detect whether a cow is sick, or to forecast cows' activities

[1] Assuming the hop-count as a metric, the reward is always equal to *1*, and the Q-value of each node represents a cost to get the content, which increases as the distance to the content source increases. The Q-value of the content source is *0*.

such as heat periods to make breeding decisions more accurate. Using the data-centric NDN security, the information related to each cow is signed directly when it is collected by the sensor (i.e. NED). Hence, every individual cow has a unique identity (not an address) and it is bound to its data at network level.

The implementation is based on a Linux PC as the gateway/local-manager (LM) and Arduino DUE boards with XBee S1 modules as NEDs. A monitoring application periodically sends Interests (44 bytes) to request content produced by NEDs (Data packet of 105 bytes).

Some measurements have been made and reported in Table II. In comparison, the RTT measured is below 6LoWPAN performance (*RTT = 9 to 25ms* [19], [20]). However, 6LoWPAN packets are not signed and additional layers are required to support data naming, while the measured RTT includes Data creation and signature, and Interest-Data (de)compression.

The compression ratio is 1.35, which is to say, 39 Bytes are gained by the packet compression in each Interest-Data exchange. The compression delay (CD) is $6\mu s$ as it only consists on adding/skipping bytes.

The implementation of the NDN802.15.4 module occupies 6% of the total memory. It includes the strategy layer with the three NDN tables and the iLNFS operations, without signature algorithms. As an empirical comparison, some open source implementations of the IPv6 stack over IEEE802.15.4 take about 12% storage and 45% RAM in the Arduino Mega, while the NDN-802.15.4 module takes approximately 5% storage and 20% RAM.

TABLE II: Deployment measurements

Metric/Operation	Value
RTT	*72 ms*
CR	*1.35*
CD	*6 μs*
First (random) Interest forwarding	*55μs*
Subsequent Interest forwarding	*10μs*
FIB entry update	*18μs*

B. Wireless forwarding performance

To evaluate the wireless forwarding strategy, we implemented it in the OMNeT++ simulator[2]. We considered a configuration with one gateway and *30* end-devices moving in an area of *200*x*200* m^2 following the random way point model, at a speed of *1 m/s* to *3 m/s*. Packet sizes are the same as in the deployment. The IEEE 802.15.4 characteristics with a communication range of *35 m* are considered.

Figure 3 reports the average energy consumption of end-devices in comparison to AODV, under the same radio consumption parameters. We chose AODV as a reference as it is a reactive protocol with a communication schema close to NDN.

[2]To accommodate non-NDN terminology, in this Subsection we refer to LM and NED by *gateway* and *end-device* respectively. Likewise, Interest and Data are referred to as *Request* and *Response* respectively.

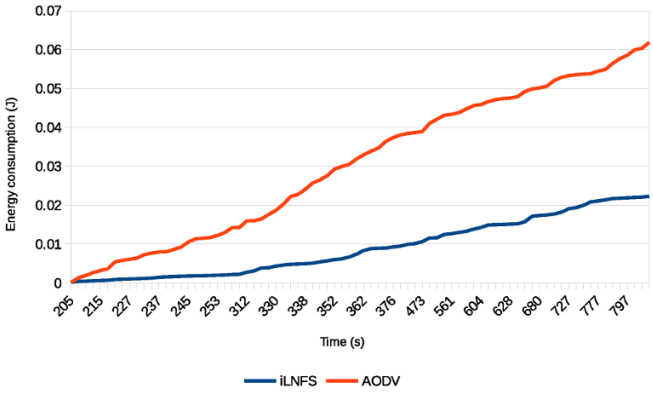

Fig. 3: iLNFS and AODV energy consumption

For *500* requests issued by the gateway, iLNFS generated 7653 ± 190 transmissions, while AODV generated 12000 ± 500. This is due to the fact that iLNFS does not use route discovery, and the flooding is reduced to the bare minimum. Moreover, as no additional packet type exists in iLNFS, the low number of transmitted packets may indicate that the best paths are frequently used to retrieve content, which is not guaranteed when the forwarding relies on Interest flooding with no route discovery. The difference in the number of transmitted packets explains why iLNFS globally consumes less than AODV.

iLNFS outperforms AODV in terms of round-trip time, mainly because of the network density in which AODV is penalized by the medium access algorithm through delayed transmissions and transmission retries.

C. Full-scale overhead

We considered a full-scale topology consisting of 6 core routers, 5 LWNs and 3 groups of consumers placed over the network.

Based on the packet exchange trace obtained for 5 minutes of simulation, we calculated the overhead generated by the network according to the number of consumers, with an estimation of UDP/IPv6 overhead with packet compression (Figure 4).

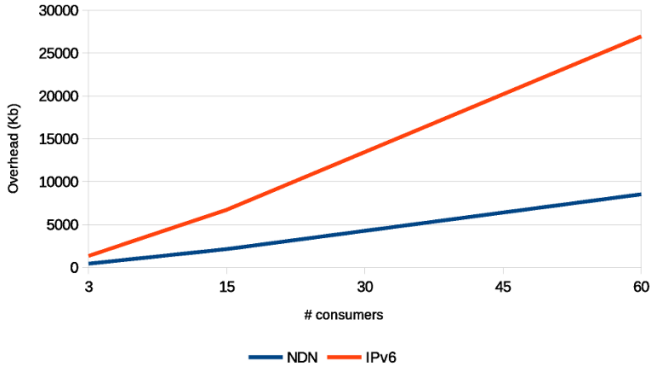

Fig. 4: Total overhead of NDN and UDP/IPv6

NDN generates much less additional data than UDP/IPv6 because of the NDN caching that reduces the number of packet

hops, helped by the NDN-802.15.4 features that save sending certain fields with packet compression, and maximize payload size with the Name-Payload-Field function.

Considering a realistic deployment of NDN over UDP/IPv6, we observe 7% to 20% overhead reduction when using the proposed NDN integration.

D. Feature comparison

A comparison between 6LoWPAN (IP) and NDN-802.15.4 (NDN) features is given in Table III.

TABLE III: NDN-802.15.4 and 6LoWPAN features comparison

Feature	NDN-802.15.4	6LoWPAN
Fragmentation	Yes	Yes
Packet structure	Name-Payload-Field balancing	Fixed packet format
Compression	Packet compression	Header compression
Mobility	Simple adaptations	Additional protocols NEMO, AdapterMIPv6, etc.
Security	Native data-centric security	MAC and TLS security

VII. Conclusion

In this paper, a key aspect of NDN for the IoT was investigated by proposing a realistic NDN-802.15.4 architecture. The main integration issues have been discussed, and the propositions were evaluated in a real-world deployment, a local-scale and full-scale simulations.

This design is a first step towards the integration of the ICN paradigm in constrained IoT environments, by giving another look at some IP-based integration techniques (e.g. 6LoWPAN). The proposed mechanisms show the flexibility of NDN for low rate lossy technologies such as the IEEE802.15.4. The NDN-802.15.4 architecture obtained aims to shape a novel and strong NDN-IoT duo.

Moreover, a lightweight NDN forwarding strategy in wireless networks was proposed. This AI-inspired mechanism provides a realistic wireless forwarding for NDN without changing its structures.

As a future work, we aim to investigate the proposed operations in depth and make an accurate performance comparison of NDN and IP architectures.

References

[1] I. Yaqoob, E. Ahmed, I. A. T. Hashem, A. I. A. Ahmed, A. Gani, M. Imran, and M. Guizani, "Internet of things architecture: Recent advances, taxonomy, requirements, and open challenges," *IEEE Wireless Communications*, vol. 24, no. 3, pp. 10–16, June 2017.

[2] IETF. Transmission of IPv6 Packets over IEEE 802.15.4 Networks. [Online]; https://tools.ietf.org/html/rfc4944.

[3] W. Shang, Y. Yu, R. Droms, and L. Zhang, "Challenges in IoT networking via TCP/IP architecture," NDN, Tech. Rep. NDN-0038, February 2016.

[4] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, kc claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, "Named Data Networking," *ACM SIG-COMM Computer Communication Review*, vol. 44, no. 3, pp. 66–77, July 2014.

[5] D. Saxena, V. Raychoudhury, N. Suri, C. Becker, and J. Cao, "Named data networking: A survey," *Computer Science Review*, vol. 19, pp. 15 – 55, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1574013715300599

[6] 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 *Proceedings of the 1st ACM Conference on Information-Centric Networking*, ser. ACM-ICN '14. New York, NY, USA: ACM, 2014, pp. 77–86. [Online]. Available: http://doi.acm.org/10.1145/2660129.2660144

[7] A. Bannis and J. Burke, "Creating a secure, integrated home network of things with named data networking," NDN, Tech. Rep. NDN-0035, November 2015. [Online]. Available: http://named-data.net/publications/techreports/ndn-0035-1-creating_secure_integrated/

[8] W. Shang, Q. Ding, A. Marianantoni, J. Burke, and L. Zhang, "Securing building management systems using named data networking," *IEEE Network Journal*, vol. 28, no. 3, pp. 50–56, 2014.

[9] J. Burke, P. Gasti, N. Nathan, and G. Tsudik, "Secure sensing over named data networking," in *2014 IEEE 13th International Symposium on Network Computing and Applications*, Aug 2014, pp. 175–180.

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

[11] Z. Ren, M. A. Hail, and H. Hellbrck, "Ccn-wsn - a lightweight, flexible content-centric networking protocol for wireless sensor networks," in *2013 IEEE Eighth International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, April 2013, pp. 123–128.

[12] W. Shang, A. Afanasyev, and L. Zhang, "The design and implementation of the ndn protocol stack for riot-os," in *2016 IEEE Globecom Workshops (GC Wkshps)*, Dec 2016, pp. 1–6.

[13] A. Abane, M. Daoui, S. Bouzefrane, and P. Muhlethaler, "Ndn-over-zigbee: A zigbee support for named data networking," *Future Generation Computer Systems*, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X17303928

[14] W. Shang, A. Bannis, T. Liang, Z. Wang, Y. Yu, A. Afanasyev, J. Thompson, J. Burke, B. Zhang, and L. Zhang, "Named data networking of things (invited paper)," in *2016 IEEE First International Conference on Internet-of-Things Design and Implementation (IoTDI)*, April 2016, pp. 117–128.

[15] J. Shi, E. Newberry, and B. Zhang, "On broadcast-based self-learning in named data networking," in *2017 IFIP Networking Conference (IFIP Networking) and Workshops*, June 2017, pp. 1–9.

[16] L. Wang, A. Afanasyev, R. Kuntz, R. Vuyyuru, R. Wakikawa, and L. Zhang, "Rapid traffic information dissemination using named data," in *Proceedings of the 1st ACM Workshop on Emerging Name-Oriented Mobile Networking Design - Architecture, Algorithms, and Applications*, ser. NoM '12. New York, NY, USA: ACM, 2012, pp. 7–12. [Online]. Available: http://doi.acm.org/10.1145/2248361.2248365

[17] M. Michael, P. Vasileios, and Z. Lixia, "Listen first, broadcast later: Topology-agnostic forwarding under high dynamics," in *Annual conference of international technology alliance in network and information science*, 2010.

[18] Y. T. Yu, R. B. Dilmaghani, S. Calo, M. Y. Sanadidi, and M. Gerla, "Interest propagation in named data manets," in *2013 International Conference on Computing, Networking and Communications (ICNC)*, Jan 2013, pp. 1118–1122.

[19] G. Gardasevic, S. Mijovic, A. Stajkic, and C. Buratti, "On the performance of 6lowpan through experimentation," in *2015 International Wireless Communications and Mobile Computing Conference (IWCMC)*, Aug 2015, pp. 696–701.

[20] B. Cody-Kenny, D. Guerin, D. Ennis, R. Simon Carbajo, M. Huggard, and C. Mc Goldrick, "Performance evaluation of the 6lowpan protocol on micaz and telosb motes," in *Proceedings of the 4th ACM Workshop on Performance Monitoring and Measurement of Heterogeneous Wireless and Wired Networks*, ser. PM2HW2N '09. New York, NY, USA: ACM, 2009, pp. 25–30. [Online]. Available: http://doi.acm.org/10.1145/1641913.1641917