Named Data Networking: a Natural Design for Data Collection in Wireless Sensor Networks

Marica Amadeo, Claudia Campo, Antonella Molinaro, Nathalie Mitton

Named Data Networking: a Natural Design for Data Collection in Wireless Sensor Networks

Marica Amadeo*, Claudia Campolo*, Antonella Molinaro*, Nathalie Mitton†
*University Mediterranea of Reggio Calabria, Italy, E-mail: name.surname@unirc.it
†INRIA Lille-Nord Europe, 59650; France, E-mail: nathalie.mitton@inria.fr

Abstract—**Named Data Networking (NDN) is a promising paradigm for the future Internet architecture that also opens new perspectives in the way data can be retrieved in Wireless Sensor Networks (WSNs). In this paper, we explore the potentialities of the NDN paradigm applied to WSNs and propose enhancements to the NDN forwarding strategy by including principles inspired by traditional data-centric routing schemes. Results achieved through the ndnSIM simulator confirm the viability and effectiveness of the proposal.**

Index Terms—**Named Data Networking, Wireless Sensor Networks, Directed Diffusion**

I. INTRODUCTION

The rapid deployment of low-cost sensing devices is offering unprecedented opportunities to the creation of highly instrumented environments and is opening new frontiers in a variety of applications based on data collection and dissemination, such as environmental control (e.g., for temperature and pollution traces in smart cities), energy usage monitoring (e.g., for personalized energy management in smart homes), and so on.

The desire of featuring sensing devices at a global scale have pushed towards connecting every sensor to the Internet, according to the Internet of Things (IoT) paradigm [1].

In parallel, the increasing convergence demands among heterogeneous networks and the need for robust and efficient information dissemination in both wired and wireless networks have pushed the research community to explore groundbreaking networking paradigms for the future Internet.

The newly proposed Named Data Networking (NDN) architecture [2] (aka Content Centric Networking [3]) is among these efforts. NDN shares some principles with data-centric approaches, proposed since the early 2000s for WSNs [4]; the basic idea is to decouple the sensed data from the node identity.

However, so far, data-centric approaches have been mainly considered as stand-alone solutions for routing, designed to accomplish well-specific tasks without interaction with the Internet. Moreover, the problems of naming, privacy and security have been considered in isolation, without any reference to a global standard architecture.

Compared to traditional data-centric routing protocols (e.g., directed diffusion [5]), NDN proposes a comprehensive architecture for receiver-driven named data retrieval. It has been

successfully applied in challenged ad hoc wireless segments (e.g., in [6], [7] and references therein).

But only a few efforts have been devoted to evaluate NDN in WSNs. Security and naming issues are addressed in [8] and [9]. In [10], a NDN communication stack is integrated into Contiki and evaluated on a real WSN deployment. There is a lack of work focusing on the design of improved forwarding routines; this work aims to fill this gap and provides the following contributions:

1. we summarize the benefits of NDN in WSNs and discuss its potentials compared to traditional data-centric routing (Sections II and III);

2. we first present a simple data retrieval scheme that leverages the basic NDN forwarding fabric enhanced with packet overhearing to reduce collisions and duplicated transmissions over the shared medium; and then we extend the basic NDN forwarding fabric with some principles inspired by the data-centric directed diffusion routing technique (Section IV);

3. we implement our solution in ndnSIM [11], the official recently developed NDN simulation module for the Network Simulator-3 and we evaluate performance results (Section V).

II. BACKGROUND

NDN proposes a simple communication model based on the exchange of two packet types: the *Interest* and the *Data* that carry URI-like content names.

A consumer requests a content by broadcasting an Interest packet, which carries the name of the content, over the available network interfaces. The Interest can be forwarded in the network until a provider, i.e., the original data producer or any node that maintains a cached copy, replies with Data.

The Interest processing leverages three data structures maintained in every NDN node: *(i)* the Content Store (CS) that caches incoming Data, *(ii)* the Pending Interest Table (PIT) that keeps track of the forwarded Interests, and *(iii)* the Forwarding Information Base (FIB), populated by a specific routing protocol, used to relay Interests towards content source(s).

When an Interest arrives, a NDN node runs the following algorithm:

- if it has a Data packet in the CS that matches the Interest, it transmits the packet on the same interface the Interest arrived from;

- otherwise, if there is an exactly-matching PIT entry, the Interest's new arrival interface is added in the PIT entry and the packet is discarded;

- otherwise, if there is a matching FIB entry, the Interest is forwarded on the outgoing interface(s) specified in the entry, and a new PIT entry is created for the Interest.

Data packets follow the chain of PIT entries back to the original requester(s).

The so-called Strategy Layer in the envisioned NDN hourglass model manages the receiver-driven transport solutions, including the scheduling of Interest retransmission on a particular interface and the priority selection for different Interests.

NDN supports security by making digital signatures mandatory for all contents. It does not mandate any particular certification infrastructure, relegating trust management to individual applications. Private content is protected with encryption performed by the content publisher.

A. Benefits in WSNs

NDN may represent a valid solution for WSNs. Indeed its features match the use cases and applications developed on top of sensors and well cope with their potential constraints. Specifically, NDN can offer:

Easy data retrieval. Hierarchical naming facilitates content search and retrieval in large-scale WSNs and could make data aggregation easier.

Scalability. The paradigm is particularly indicated to retrieve data from several nodes in a monitored area. By leveraging the broadcast wireless medium and enforcing lightweight forwarding procedures, NDN prevents degradation of the communication quality as the number of involved nodes increases.

Caching. The cached content at different nodes (according to their storage capability) can be available also under intermittent connectivity (e.g., due to low-power operation).

Easy application development. Applications can ask for data in a content-centric manner regardless of the data's physical location, which therefore is transparent to applications.

Deployment flexibility. NDN can be implemented either as a *clean-slate* solution replacing TCP/IP protocols on top of layer 2 technologies, or as an *overlay* on top of an IP network to allow backward compliance with the existing core network. Hybrid approaches can be also deployed with a wired segment where IP is largely used and a data-centric WSN in the access segment.

III. DIRECTED DIFFUSION VS. NDN

Applications and routing requirements for WSNs are significantly different from other wireless networks. The typical communication mode is from multiple data sources to a sink, rather than between any pair of nodes. Moreover, the collected data is more important than the identity of the sensing device. Hence, researchers have found it useful to adopt data-centric abstractions to address the sensed information instead of the sensor.

Directed diffusion (DD) was proposed as a robust data-centric dissemination protocol for WSNs in [5]; data generated by sensors are named by attribute-value pairs and all nodes are application-aware. DD proposes a pull-based service that starts when the sink node *floods* Interest packets containing the attributes of the required Data. At the Interest reception, intermediate nodes set up a *gradient*, i.e., a *direction state* in a local cache that identifies which neighbour sent the Interest, and, then, they re-broadcast the request. Similarly to PIT entries in NDN, the gradient is used to set a reverse link for Data forwarding and it is essential to suppress duplicate messages and prevent loops.

When the Interest is received by a data producer, it broadcasts an *exploratory* Data packet to setup the path towards the sink. Every node receiving the Data checks in the local cache if it has received the same message before. If a Data item exists, the node drops the message. Otherwise, it looks for a pending Interest and, if a match is found, the Data is forwarded, while a copy is temporarily locally stored.

On receiving the exploratory Data, the sink sends positive or negative gradient reinforcement messages to respectively select or prune parts of the path on the basis of some performance metrics, e.g., the lowest latency. The neighbours in turn reinforce their preferred upstream next hop, according to the same metric. Future data messages from the sensor sources only travel down these reinforced gradients.

Similarly to NDN, DD performs hop-by-hop communication without following the traditional end-to-end paradigm so that, in both cases, nodes do not need to have globally unique network addresses. However, two main differences can be considered concerning the basic networking aspect:

1) DD nodes do need to distinguish between neighbours. Conversely, with NDN the knowledge of the neighbours identifiers is not necessary, since the FIBs can be populated by using a routing protocol that considers only data names and network interfaces.

2) In DD, once a path is set up between a sink and sensors, more Data packets can be forwarded towards the sink. The Interest, in fact, carries an *Interval* attribute that specifies the frequency of data sending and a *Duration* attribute that advertises how long the Interest can be maintained in the local cache. NDN, instead, assumes a pure receiver-driven logic in which *one Interest is consumed by one Data packet*. This ensures flow balance in the network and also the absence of unsolicited traffic but it may not well suit the traffic patterns of some WSN applications. The novel notion of *long-lived Interests* in [12] could be more appropriate and requires further investigation.

IV. SENSOR NETWORKING WITH EXTENDED NDN

A. Reference scenario and assumptions

In this section, we describe the proposed NDN forwarding extension with specific reference to a WSN that supports monitoring tasks via multihop communications among static nodes. The sink periodically retrieves data from the deployed sensors. The monitored region could be a single building, or a residential district in a smart city where an automation system controls some parameters (e.g., temperature, humidity, pollution, energy consumption) by periodically broadcasting Interest packets. Data collection can serve the purposes of making statistics and event prediction and let the system take

a decision e.g., turn on/off the air conditioning in some rooms, limit vehicle traffic, etc.

We assume that sensor nodes can cache Data. This is reasonable given the typical small footprint of collected data (nearly 100 bytes per Data packet).

Communication is based on Interests and Data broadcasting that follows the NDN framework enhanced with a customized naming and forwarding schemes. Specifically, two solutions have been designed: *(i)* the *basic NDN* (shortened as b-NDN) is the original NDN (with the same Interest and Data processing, packet format, and data structures) that is straightforwardly extended to cope with collisions and packet redundancy in a wireless environment, and *(ii)* the *directed diffusion NDN* (shortened as dd-NDN) that extends the basic NDN delivery with DD principles [5] to better support periodic monitoring in WSNs.

B. Naming Scheme

The designed naming scheme is common to both b-NDN and dd-NDN. In sensor networking, the naming system must accurately describe the sensing task thus allowing *(i)* the sink to fully express what information it needs and *(ii)* the sensors to precisely describe the sensed data. Therefore, the data naming design should include the following attributes:

- *Task type.* Name of the sensing task, e.g., temperature, humidity.

- *Location.* Identifier of the geographic area in which the task is performed. It can have different granularities, from hundreds to a few meters. Depending on the application scenario, such attribute can include GPS coordinates of a region or logical names, e.g., in a building, the rooms could have specific unique names such as *room-Alice*, or *room-10*.

- *Task time.* Time period in which the task is performed. It can be a real-time value (e.g., the temperature sensed when the Interest is received), or an averaged value over a given time interval (e.g., minutes or hours). In fact, NDN can support a mix of statically cached and dynamically-generated content [3], allowing a sensor to generate data on the fly in response to an Interest.

- *Duplication detection.* Since more sensors may answer to an Interest, the sink must identify data replicas. This can be accomplished by adding a random nonce to every data name[1].

The user-friendly hierarchical naming scheme of NDN easily matches all the above mentioned attributes. Specifically, we assume the following names structure: *"task type"/"task location"/"time period"/nonce*. For instance, a Data packet with name *"temperature/room10/timestamp/12132343"* declares that it carries a temperature measurement, taken in room10 at the time indicated by the timestamp. Similarly, an Interest with name *"humidity/area121/[timestamp1, timestamp2]/1323454"* declares that the sink is looking for an humidity average data, taken in area121 in the time period between *timestamp1* and *timestamp2*.

---

[1]In the standard NDN, Interests already include a nonce value to discard duplicates, while Data packets do not require nonce values.

The hierarchical structure of the NDN naming allows to perform data retrieval at different granularities. For instance, the lack of the task time attribute in an Interest *"humidity/area121/1323454"* means that the sink is looking for humidity data in area 121, regardless of the time of the sensed task.

C. Basic NDN Operation

The legacy NDN assumes that an Interest is forwarded over some available network interfaces except from the one the Interest arrived from. In our scenario, each sensor is provided with a single radio interface; as a consequence, a first required modification is *to forward packets over the same (unique) network interface they arrived from*. Specifically, we define an Interest rebroadcasting routine with a twofold objective: *(i)* minimizing collisions between forwarded Interest and Data packets and *(ii)* reducing packets redundancy.

Here, we assume that Interest and Data rebroadcasting events are deferred of $T_{Interest}$ and T_{Data} time values, respectively. Both $T_{Interest}$ and T_{Data} are randomly computed by considering the so-called *defer window* (DW), an integer value that indicates the length of the time intervals. The values are calculated as follows:

$$T_{Data} = rand[0, DW] * DeferSlotTime$$
$$T_{Interest} = (DW + rand[0, DW]) * DeferSlotTime \quad (1)$$

where $DeferSlotTime$ is a fixed, short time interval. As a consequence, $T_{Interest}$ and T_{Data} are selected in disjoint intervals with $T_{Interest} > T_{Data}$ to give higher access priority to Data packets.

During the $T_{Interest}$ waiting time, a potential Interest forwarder listens to the channel: if it overhears the same Interest or the requested Data, then it cancels its own transmission and deletes the related PIT entry. Such an approach is more efficient than a naive flooding, as for instance the one deployed for Interest propagation in DD.

The sink node retransmits the Interest if the related Data packet is not received within a given time interval. At this stage of research, we set the Interest retransmission interval to a fixed value, since latency requirements are not so severe and the monitoring system can tolerate variable delays up to a few seconds.

All in all, by using such a simple broadcast protocol, a sink can collect data without any knowledge about the sensor identifiers or the network topology. However, some improvements can be conceived, as detailed in the next subsection, to better match the targeted applications.

D. NDN enhanced with Directed Diffusion principles

In the envisioned static scenario, once the sink has transmitted the first Interest and retrieved the relevant Data, a path has been discovered that likely will last long. Therefore, the NDN forwarding strategy could benefit from maintaining some information about the discovered path to be used for subsequent Interests. This would reduce traffic overhead and

(a) Interest broadcasting by the sink.　　(b) Interest forwarding by intermediate nodes.　　(c) Path establishment.
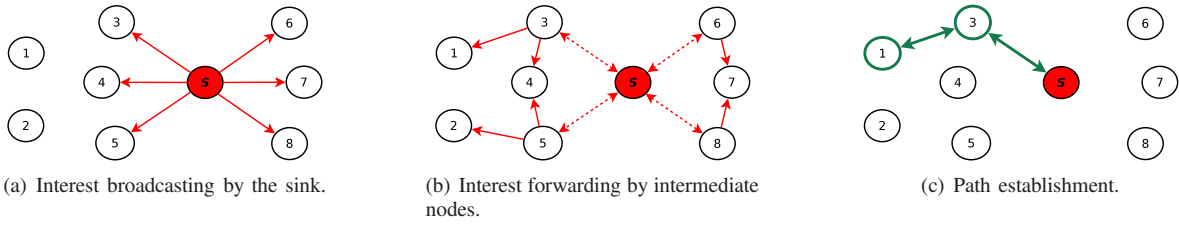
Fig. 1.　Example of path establishment for dd-NDN.

collisions by limiting the number of sensors involved in the delivery process. To fulfil this objective, a forwarding strategy has been designed in dd-NDN that leverages the b-NDN fabric further enhanced with a *direction state* that is set during the Data packet forwarding in the initial discovery phase, similarly to DD.

Specifically, when the first Interest reaches the data producer, it includes its *identifier*[2] in the Data packet. On receiving the Data, a node that has a corresponding PIT entry (and therefore can forward the Data towards the sink) includes the previous node identifier in a new table called *Next Hop Table* (NHT). A NHT entry contains a bind between the data name declared in the Interest/Data packet (except for the *nonce* value) and the next hop identifier; it does not contain information about the entire path. If different Data are received from the same next hop, the corresponding data names can be aggregated in the same entry. This allows for scalability with respect to the number of monitoring tasks. Moreover, the NDN forwarding fabric inherently discards duplicated Data, since one Interest is consumed by only one Data. Therefore, only a next hop is considered for a specific Data.

Every node that can forward a Data packet maintains a corresponding NHT entry, including the sink. Thus, a path is created between the sensor that originated the data and the sink. When the sink sends a subsequent Interest to retrieve the same data type, it includes the identifier of its immediate next hop thus allowing only that node to forward the packet. In case of failure along the path, a Data packet can be retrieved once a new Interest is issued by the sink and the recovery is speeded up thanks to caching at intermediate nodes. Furthermore, if a selected next hop N does not reply to Interests after a fixed number of retry times, the sender assumes that N is unreachable (e.g., it ran out of energy), and it starts a new discovery phase.

To better explain our proposed solution, we consider the example in Fig. 1. The sink broadcasts an Interest with name */humidity/area1* that is received by all its neighbours, nodes 3-8 in Fig. 1(a). No node has the requested Data, so the Interest is rebroadcasted after a random defer time, $T_{Interest}$.

In Fig. 1(b), nodes 3, 5, 6 and 8 seize the channel and transmit the packet, while nodes 4 and 7 receive the duplicated Interest from their neighbours and cancel their own transmission. Nodes 6 and 8 do not receive any answer to the Interest

and finally discard the corresponding PIT entry. Instead, node 3 receives the Data from node 1 and includes in the NHT node 1 for Data */humidity/area1*. Then, it forwards the packet towards the sink that, in turn, includes node 3 in its NHT, Fig. 1(c). When the sink transmits the subsequent Interest for */humidity/area1*, it advertises node 3 as next hop forwarder and all the other nodes will simply discard the packet. Node 3, in turn, advertises node 1 when it forwards the Interest.

V. PERFORMANCE EVALUATION

To evaluate our NDN solution for WSNs we use the Network Simulator 3 (ns3) and the open-source ndnSIM [11] module, specifically designed to support NDN networks.

We consider a 400m x 400m area with 80 sensors deployed in a lattice topology, and a sink node in the middle of the topology (Fig. 2). Each node uses IEEE 802.11g radio interface[3]; physical and medium access control parameters are set according to [13], a small form factor, ultra-low power networking module for home automation, remote equipment monitoring, etc. The coverage range of each node is about 80 m and the distance between two adjacent sensors is 50 m. Therefore, the number of links that every node can establish is eight, with the exception of the nodes at the border that can communicate with 3 or 5 neighbours.

We extended ndnSIM in order to support the new forwarding strategies, and specifically:

- packet forwarding over the same 802.11 radio interface;
- packet suppression based on overhearing and defer times;
- forwarding strategy inspired by the DD scheme and relying on the new packet field with node identifier and the new *NHT* data structure.

We assume that the sink broadcasts Interests for a specific task (i.e., data to be retrieved) at fixed time instants. Specifically, we consider a monitoring interval of 60s. We assume that the sink sends 4 different Interests per monitoring task, n_t, whose number varies from 1 to 16. Therefore, if the data to be monitored is only one (e.g. temperature/area0/now), the sink sends the first Interest at t=0, the second Interest at t=60, and so on. If the data to be monitored are four (e.g., temperature/area0/now, temperature/area10/now, temperature/area20/now, temperature/area30/now), the sink sends four corresponding Interests at t=0, four Interest at t=60 and so on. In case the Data is not received within a timeout interval,

---

[2]Different locally unique node identifiers may be used, e.g., MAC addresses or logical names. This information is included in a new field of Interest/Data packets' headers.

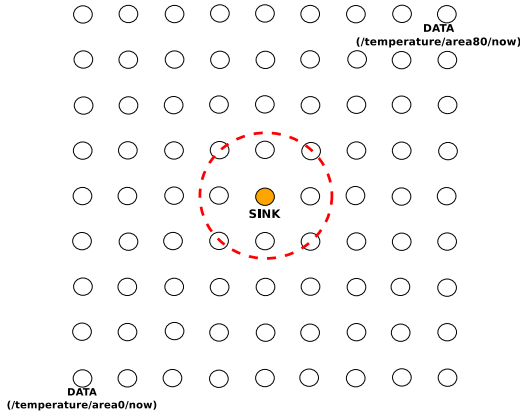[3]The Wi-Fi Alliance is rolling out lower-power chipsets that meet battery life requirements for sensors.

Fig. 2. Simulation Scenario: lattice topology.

TABLE I
SIMULATION PARAMETERS.

b-NDN Parameter	Value
DW	15, 31, 127, 255, 511, 1023
DeferSlotTime	28 μsec (802.11g DIFS)
Interest retransmission timeout	1s
Application Parameter	**Value**
Monitoring tasks	1, 4, 8, 12, 16
Monitoring interval	60s
Interest per tasks	4
Data payload	100 bytes
Scenario Parameter	**Value**
Simulation area	400m x 400m
Topology	Lattice with step 50m
Sink/Sensors	1/80

the Interest is retransmitted by the sink. Simulations end when all the required Data packets are retrieved by the sink.

First, we analyze the b-NDN proposal and study the impact of different defer window values in the range DW=[15, 1023], when varying the number of monitoring tasks. Once found the DW value that achieves the best trade-off between efficiency and effectiveness in data retrieval, the b-NDN and dd-NDN are compared in terms of: *(i) data retrieval delay*, computed as the time required to perform the monitoring task, since the sink transmits the Interest to the reception of the requested Data; and *(ii) Interest and Data overhead*, computed, respectively, as the number of Interest and Data packets transmitted in the network (by all sensors and the sink) over the number of Interests sent by the sink (without counting retransmissions).
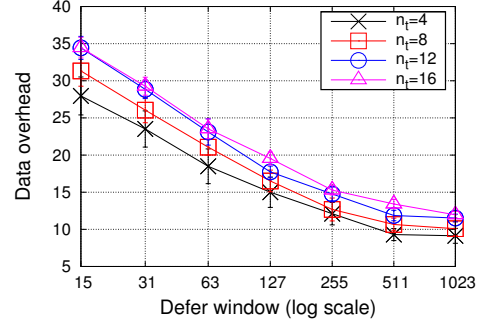
The main simulation parameters are summarized in Table I. Simulation results are averaged over 20 independent runs and reported with the 95% confidence intervals.
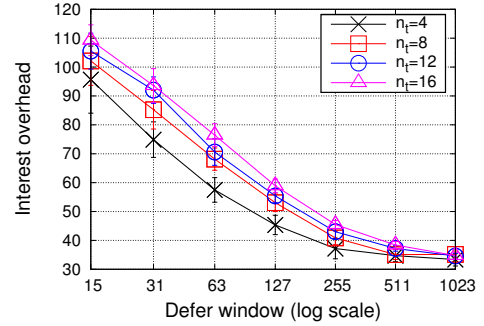
A. Basic NDN

Results in Fig. 3(a) show the impact of the defer time on the data retrieval delay. The latter one gets significantly shorter when the DW size passes from 15 to 511 due to the fact that larger DW sizes correspond to less likely collisions and, consequently, speed up data retrieval from sensors. This also implies that the proposed broadcasting technique with collision avoidance timers can be more effective than a naive Interest

(a) Data retrieval delay.

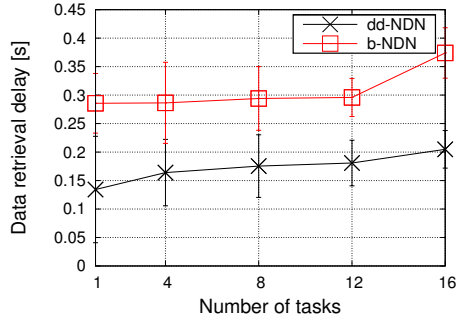(b) Data overhead.

(c) Interest overhead.

Fig. 3. Metrics for b-NDN when varying the Defer Window (DW) and the number of monitoring tasks (n_t).

flooding. As expected, the higher the number of monitoring tasks the higher the delay. Notwithstanding, slight differences are noticed, especially when DW approaches 511. The main reason behind such a result is the light load on the channel: data are collected at a low frequency (at every 60 s) and the defer times for packets rebroadcasting are long. Hence, procedures enforced to retrieve different data in the network do not heavily compete among each other.
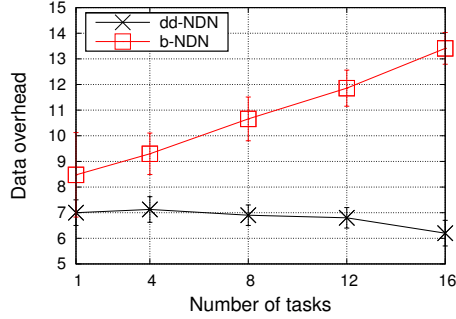
The positive effect of large DW sizes can be also noticed on Data and Interest overhead in Fig. 3(b) and (c), respectively. It is worth noticing that the Data overhead is significantly lower than the Interest overhead. In fact, PIT entries are used for restricting Data forwarding to a subset of nodes.
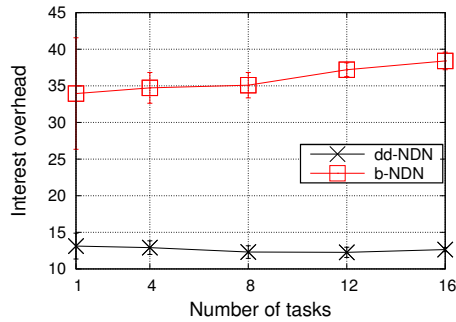
B. Directed Diffusion NDN

To compare b-NDN and dd-NDN, we set the DW to 511, since this value proved to achieve the best trade-off

(a) Data retrieval delay.

(b) Data overhead.

(c) Interest overhead.

Fig. 4. Comparison between b-NDN and dd-NDN when varying the number of monitoring tasks (n_t), DW=511.

between efficiency (overhead) and effectiveness (delay) in data retrieval. Results are reported in Fig. 4.

The dd-NDN solution can achieve data retrieval delay values that are more than halved compared to b-NDN, thanks to the preliminary data discovery phase that allows transmitting subsequent Interest/Data packets over predetermined paths. Fig. 5 shows the reduction in the retrieval delay, once the path is established compared to the delay experienced by the first received Data packet.

Thanks to path set-up during the retrieval of the first Data packet, dd-NDN also achieves significant reduction (values are almost halved) in Data and Interest overhead compared to b-NDN. Such a reduction is stronger for the Interest overhead w.r.t. the Data overhead thanks to the fact that dd-NDN uses the information stored in the NHT for successive Interest forwarding.
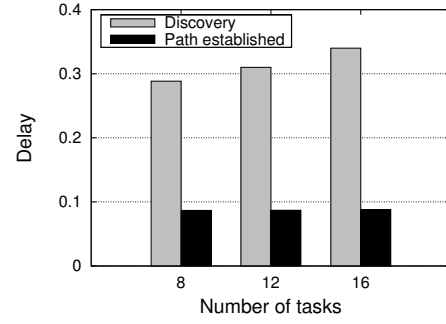

Fig. 5. Comparison between the data retrieval delay in the discovery stage and the delay after the path establishment in dd-NDN.

VI. CONCLUSION

In this paper we have investigated the named data networking paradigm in wireless sensor networks. To the best of our knowledge, this is the first work that explores NDN potentialies in WSNs and validates its performance via the official NDN open-source simulator, ndnSIM. Achieved results confirm that NDN is a promising candidate technology for WSNs and that the proposed NDN enhancements, inspired by directed diffusion and coupled with the anti-collision timers, are effective and efficient and fit well the requirements of the considered periodical monitoring applications.

Despite such merits, research about NDN in WSNs is still in its infancy. Future work will be devoted to: *(i)* evaluate performance on a wider scale, under different scenarios and applications; *(ii)* get insights into the energy efficiency of the designed solutions and specify possible improvements, e.g., to account for the sleep mode of 802.15.4 nodes.

REFERENCES

[1] M. Palattella *et al.*, "Standardized protocol stack for the Internet of (important) Things," *IEEE Communications Surveys & Tutorials*, 2013.
[2] "Named Data Networking (NDN) Project, http://www.named-data.net/."
[3] V. Jacobson *et al.*, "Networking Named Content," in *Proc. ACM CoNEXT*, 2009.
[4] T. Watteyne, A. Molinaro, M. G. Richichi, and M. Dohler, "From MANET to IETF ROLL standardization: A paradigm shift in WSN routing protocols," *IEEE Communications Surveys & Tutorials*, vol. 13, no. 4, pp. 688–707, 2011.
[5] C. Intanagonwiwat *et al.*, "Directed Diffusion for Wireless Sensor Networking," *IEEE/ACM Trans. on Networking*, vol. 11, no. 1, 2003.
[6] M. Amadeo, A. Molinaro, and G. Ruggeri, "E-CHANET: Routing, Forwarding and Transport in Information-Centric Multihop Wireless Networks," *Computer Comm.*, vol. 36, no. 7, pp. 792–803, 2013.
[7] M. Amadeo, C. Campolo, and A. Molinaro, "CRoWN: Content-Centric Networking in Vehicular Ad Hoc Networks," *IEEE Communications Letters*, vol. 16, no. 9, September 2012.
[8] L. Breslau *et al.*, "Securing Instrumented Environments over Content-Centric Networking: the Case of Lighting Control and NDN," in *IEEE INFOCOM NOMEN Workshop*, 2013.
[9] J. Zhang *et al.*, "iHEMS: An information-centric approach to secure home energy management," in *IEEE SmartGridComm*, 2012.
[10] B. Saadallah, A. Lahmadi, and O. Festor, "CCNx for Contiki: Implementation Details," INRIA, Tech. Rep. RT-0432, November 2012.
[11] A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnSIM: NDN simulator for NS-3," NDN Project, Tech. Rep. NDN-0005, July 2012.
[12] A. Carzaniga *et al.*, "Content-Based Publish/Subscribe Networking and Information-Centric Networking," in *ACM SIGCOMM ICN 2011*.
[13] "RN-171 802.11 b/g Wireless LAN Module, Rowing Networks," Datasheet available on line at: http://www.microchip.com/.