Cluster Computing (2019) 22:805–818
https://doi.org/10.1007/s10586-018-2859-7

Efficient forwarding strategy in a NDN-based internet of things

Ahmed Aboud[1,2] · Haifa Touati[1] · Brahim Hnich[3,4]

Received: 29 October 2017 / Revised: 23 September 2018 / Accepted: 19 October 2018 / Published online: 30 October 2018

Abstract

The future internet will include a variety of heterogeneous wireless networks and all kinds of smart sensors connected over wireless links. The size and cost constraints on sensor nodes result in corresponding constraints on resources such as energy, memory, computational speed and communications bandwidth which in turn limits the processing and communication capabilities of the sensor nodes. Named data networking (NDN) emerged as a promising new routing mechanism aimed to cope with the increasing number of heterogeneous networks and the need for efficient and robust data dissemination. But, one of the bottlenecks that hinders the full applicability of such a NDN-based approach to the future internet is the energy requirements. In this study, we propose a novel geographic interest forwarding scheme where we add support for push-based traffic and different forwarding techniques designed to balance the energy consumption across the network. Our simulation results show that our proposed approach is more scalable and outperforms the alternative methods in term of data retrieval delay and overall energy consumption.

Keywords Named data networking · Internet of things · Wireless sensor networks · Multihop communications · Performance evaluation

1 Introduction

The internet of things (IoT) has emerged as a new technology that is gaining an increasing popularity in recent years for interconnecting different devices such as sensors, equipments and systems [1]. The future internet will include a variety of heterogeneous wireless networks and all kinds of smart objects connected over wireless links. These objects consists of small, low-cost, low-power, multifunctional, autonomous devices. These sensors are used to sense the ambient condition of its surroundings, gather *Data*, and process it to draw some meaningful information about its environment [2].

The size and cost constraints on sensor nodes result in corresponding constraints on resources such as energy, memory, computational speed and communications bandwidth which in turn limits the processing and communication capabilities of the sensor nodes. In this context, it is essential that solutions deployed on top of an IoT network must be cooperative and make use of creative distributed techniques both in the communication mode as well as the processing of the acquired information.

The idea of the IoT started as a futuristic concept and has come a long way. But, according to the Internet of Things (IoT) paradigm [3], there is an increasing desire of featuring sensing devices at a global scale by connecting every sensor to the Internet. Hence, there exists a research urge to explore the design of new routing mechanisms due to the increasing number of heterogeneous networks and the need for efficient and robust Data dissemination in such a Future Internet.

The newly emerging Internet architecture named data networking (NDN) [4] is among these efforts. NDN has

✉ Ahmed Aboud
ahmedaboud@outlook.com

Haifa Touati
haifa.touati@cristal.rnu.tn

Brahim Hnich
hnich.brahim@gmail.com

[1] Hatem Bettahar IResCoMath Research Unit, University of Gabes, Gabès, Tunisia

[2] National Engineering School, University of Sfax, Sfax, Tunisia

[3] CES-Laboratory, University of Sfax, Sfax, Tunisia

[4] Faculty of Sciences, University of Monastir, Monastir, Tunisia

been introduced by Van Jacobson et al. in [5] to cope with the evolution of the Internet usage towards massive content distribution and it shares some principles with *Data*-centric approaches, proposed since the early 2000s for wireless sensor networks (WSNs) [6]. Compared to traditional *Data*-centric routing schemes which are based on standalone solutions for routing that lacks the support of standard naming schemes, NDN offers a comprehensive solution for the whole Internet and the IoT.

NDN adopts a receiver-based service model and changes the communication paradigm from traditional host-centric communication to named *Data*-centric communication which naturally meets the requirements of wireless sensor networks (WSN) and IoT well.

NDN architecture deployment has been widely studied in the past in the context of wired networks but only recently in the context of the IoT. In fact, a few efforts have been devoted to the research on how to apply the NDN to the IoT and WSNs [7–12], which open doors to new challenges such as naming schemes, security, routing, and forwarding, since original NDN design is aiming at a whole Internet architecture, rather than the Internet of Things. One of the bottlenecks that hinders the full applicability of such a NDN-based approach to the future internet, is the energy requirements and the need to minimize the resource usage in such type of networks. In this study, we propose a novel geographic interest forwarding scheme where we add support for push-based traffic and different forwarding techniques designed to balance the energy consumption across the network. Our simulation results show that our proposed approach is more scalable and outperforms the alternative methods in term of data retrieval delay and overall energy consumption. The rest of the paper is organised as follows. In Sect. 2, we review the related works and provide a short description of the NDN architecture in order to highlight the most important architectural features useful in the context of our contribution. Then, we highlight our assumptions about the system model we consider in this paper in Sect. 3. In Sect. 4, we propose a NDN-based IoT forwarding extension in which we provide support for unsolicited *Data* pushing. Next, we detail our simulation model in Sect. 5 and evaluate the proposed protocol in term of energy usage, network lifetime, scalability and efficiency in Sect. 6. Finally, we conclude and point out to possible future research directions in Sect. 7.

2 Related work

2.1 NDN communication model

The NDN communication model is based on the exchange of two packet types: the *Interest* and the *Data* identified by a URI-like content names [4]. To receive *Data*, a NDN node sends out an *Interest* packet which carries a name that identifies the desired *Data* and broadcasts it over the available network interfaces. The contents are divided into *chunks*. Each chunk typically has the size of an IP packet. Each *Interest* packet matches only one *Data* packet and each *Data* packet corresponds to one *Chunk* [13].

In the following, we will name the node that sends *Interest* a *“Consumer”* and the original *Data* source a *“Producer”*. The *Interest* processing as shown in Fig. 1 leverages three *Data* structures maintained in every NDN node [5] as illustrated in Fig. 2:

- a **Forwarding *Interest* Base (FIB)** indexed by the content name prefixes and used to relay *Interest*s towards the content source(s);
- a **Pending *Interest* Table (PIT)** that keeps track of the forwarded *Interest*s;
- a **Content Store (CS)** that caches incoming *Data* packets to satisfy future requests for the same *Data*.

When a NDN node receives an Interest packet, it makes a lookup on its CS. If it has the desired Data, it transmits the Data packet on the same interface the Interest arrived from. If the match in the CS fails, the Interest packet will be stored in the PIT, where each entry contains the name of the Interest and a set of interfaces from which the matching Interests have been received. Then, the Interest will be further forwarded to the upstream neighbor(s) based on the FIB information. When a Data packet arrives, an NDN node finds the matching PIT entry and forwards the data to all downstream interfaces listed in that PIT entry. It then removes that PIT entry, and caches the Data in the Content

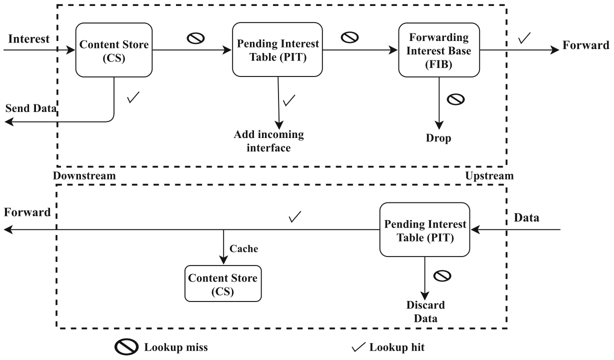
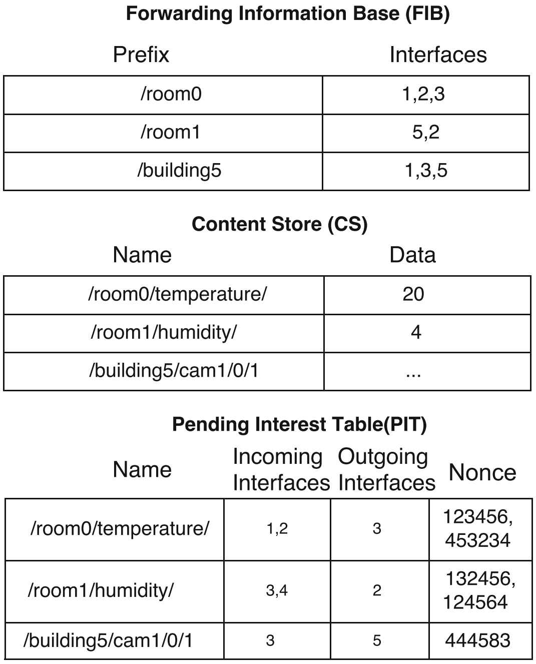


Fig. 1 NDN interest processing



Forwarding Information Base (FIB)

Prefix	Interfaces
/room0	1,2,3
/room1	5,2
/building5	1,3,5

Content Store (CS)

Name	Data
/room0/temperature/	20
/room1/humidity/	4
/building5/cam1/0/1	...

Pending Interest Table(PIT)

Name	Incoming Interfaces	Outgoing Interfaces	Nonce
/room0/temperature/	1,2	3	123456, 453234
/room1/humidity/	3,4	2	132456, 124564
/building5/cam1/0/1	3	5	444583

Fig. 2 NDN data structure

Store. If no matching PIT entry was found, the *Data* packet will be discarded.

2.2 Interest forwarding in NDN-based WSNs

NDN may represent a valid solution for the IoT since the receiver-based service model in the NDN fits naturally into *Data*-centric IoT. NDN-based IoT and WSNs are the focus of a growing research effort [7–10].

2.2.1 DD-NDN protocol

One of the very first works that explore the potential application of the NDN paradigm to WSN is inspired by the **Directed Diffusion** (DD) protocol [14] and is called **DD-NDN** [7]. The authors of DD-NDN design a blind flooding based on a deferred timer to discover potential content providers. Each node stores information about the next-hop discovered during the blind flooding phase to guide future *Interest* forwarding.

In the first phase of this protocol the sink node blindly broadcasts the *Interest* packet to all the sensor nodes in its range. When a node receives the *Interest*, it computes an *Interest* differ timer, if the receiving node does not overhear the same *Interest* from its neighbor and the *Interest* differ timer has elapsed, it will forward the *Interest* otherwise it will be dropped. When the *Interest* packet reaches a content provider, the returned *Data* packet will include an additional field that carries the identifier, *ID*, of the node that forwarded the *Data* packet and each node that receives the *Data* will store that *ID* in a new table called *Next Hop Table* (NHT). After the blind forwarding phase, when the sink node wants to send another *Interest* it first checks the *NHT* table to see if there is a possible next-hop for the *Interest* name prefix. If a match is found the *Interest* will include that *ID* in a new field and only the node that has that *ID* will be able to forward the *Interest* packet. Otherwise, the DD-NDN protocol will fallback to the blind forwarding algorithm.

This scheme minimizes the number of nodes that participate in the forwarding process. However, this proposal is inefficient in term of energy consumption due to the fact that with the increase in the number of consumers, blind flooding will be used regularly and the network lifetime will be decreased.

2.2.2 Provider-aware protocol

In [8, 15, 16], Amadeo et al. propose a **Provider-Aware Forwarding (PAF)** scheme for Ad-hoc networks in which blind flooding approach was used to discover possible content providers and to guide the *Interest* forwarding in a similar way as the DD-NDN protocol.

Each node in PAF stores the information about the number of hops between a local node and a content provider in a *Distance Table* called DT. The first *Interest* is disseminated in the network in a blind flooding way, when a node receives the request, it returns the *Data* packet which includes two additional fields. One field carries the ID of the Producer and the other the hop-count(distance) initialized to 1. Each node that has a PIT entry for that *Data* packets stores the ID, the name prefix of the packet and the distance to the content Producer in the DT table. When a node forwards the *Data* packet it increments the distance field by one until the *Data* packet reaches its destination.

When the Consumer sends another *Interest* it includes the ID and the distance to it in two additional fields. Each node that receives the *Interest* checks its DT table against the provider ID in the *Interest*. If a match is found, it compares the distance in the *Interest* to the distance in the DT table. If it is closer to the provider, that node will forward the packet; otherwise it will be dropped.

The PAF scheme has the advantages of reducing the redundancy in the network, but it suffers from the same problem as the DD-NDN protocol. When the blind forwarding approach is used to discover possible content provider, packet duplication is inevitable due to the hidden terminal phenomena.

2.2.3 Single interest multiple data

Single *Interest* Multiple *Data* (SIMD) is another approach proposed by Amadeo et al. in 2014 [9]. SIMD is a retrieval framework proposed to receive *Data* from multiple wireless sources. Authors of SIMD introduce different mechanisms to ensure reliable *Data* retrieval such as packet suppression, collision avoidance, and the use of the Exclude field carried in the *Interest* packets. However, the proposed framework is only suitable for one-hop wireless scenarios.

2.2.4 Dual mode interest forwarding

Dual Mode *Interest* Forwarding (DMIF) is a recent *Interest* forwarding scheme proposed in 2016 by Gao et al. [10] for NDN-based WSNs.

DMIF combines two *Interest* forwarding modes. The Flooding Mode(FM) is used to discover potential content producers and store their ID in the Forwarding Information Base (FIB) table. Several techniques are used to improve the performance and the energy efficiency of the proposed system such as scope control. Also, the FM mode implements a broadcast storm avoidance technique by using a differ timer that depends on the energy and the distance between sensor nodes in order to maximize the network lifetime. After the discovery of the content producer with the FM mode, the Directive Mode is used to guide the *Interest* forwarding in an energy-aware way based on the lookup results at the FIB. However, according to the author of the DMIF scheme, some of the proposed approaches may not scale well in large scale networks. Also piggybacking different information such as the energy and sender-id into NDN *Data* packet will introduce an overhead which in turn will limit the size of actual *Data* that a NDN *Data* packet can carry in an 802.15.4 frame which could result in *Data* fragmentation.

3 System model

We consider an application scenario as follows: A sensor network is deployed for monitoring physical or environmental conditions. The monitored areas could be a single building or a smart city where an automation system controls some parameters (e.g., temperature, humidity, energy consumption). A number of static sensors nodes are acting as content producers by detecting the status of each target and generating sensing *Data*.

The number of deployed sinks nodes in the network varies between 1 and N. The latter act as a contents consumers by consuming *Data* sent from sensor nodes, using two transmission modes:

- *Push-based mode:* In the first case, sensor nodes periodically send *Data* without receiving a prior explicit *Interests* from the sinks.
- *Pull-based mode:* In the second transmission mode, the sinks may issue *Interest* packets to query for *Data* about a specific parameter.

Finally, we assume that the sensor nodes are stationary, are aware of their own locations and know the positions of the sink nodes.

4 Geographic interest forwarding (GIF)

IoT routing requirements and constraints are significantly different from other wireless networks. The typical communication model is to send *Data* from multiple *Data* sources to a sink rather than between any pair of nodes. The receiver-based service model in the NDN fits naturally into *Data*-centric IoT. Indeed, the NDN communication model supports pull-based traffic naturally where no *Data* is sent unless a consumer explicitly asked for it with an *Interest* packet. Such model can significantly reduce the amount of unwanted *Data* transfers. On the other hand, since sensor nodes can not initiate communication, a wide range of applications such as alarms or status changes will not be supported and cannot be natively implemented by NDN. Hence, NDN-based IoT lacks support of push-based traffic unless additional logics are implemented in the NDN Forwarding Strategy layer.

Furthermore, as sensor nodes are generally battery-powered devices, the critical aspects to face is how to extend the network lifetime by reducing the energy consumption of nodes. On the contrary to the sink nodes, which are a powerful devices, the sensors nodes are resource limited devices in term of energy, memory and computing power requiring that the process of *Data* retrieval should be energy efficient. Hence, we propose an NDN-based Geographic *Interest* Forwarding scheme (GIF).

In the proposed GIF scheme, the NDN protocol was extended with *push-based* traffic support which will be used for two diffrent objectives:

1. For sensor nodes to transmit their data directly to the sinks nodes without requiring a prior *Interest.*
2. For the producers to announce their existance to the sink nodes.

The GIF scheme adapts a greedy forwarding strategy with energy awarness to route messages to the contents producers.

It makes uses of a neighbourhood “Hello packets” that sends a node’s identity, its position and its residual energy. However, in order to minimize the network congestion this exchange is only done once at a **Neighbor Discovery** phase. After the neighbors discovery phase, each producer will send its coordinates and the type of data they produce back to the sinks nodes using the push-based *Interest* packet in a **Producer Discovery** phase. Finally, every node in GIF has its own neighbors and producers tables populated during the first two phases. Whenever a sink node requests the data from a producer, the GIF scheme tries to find the closest neighbor to the destination that has more residual energy and forwards the message to that node. We call this phase a **Data Exchange** phase. In order to maintain the neighbor table for each node while minimizing the power consumption, GIF piggybacks the energy information on every*Interest* packet that is sent, forwarded or received by sensor nodes instead of periodically exchanging those information between neighbors. The basic modules in the GIF scheme are ***Data*** **Naming**, **Neighbor Discovery**, **Producer Discovery** and **Data Exchange**, which will be explained in the following subsections.

4.1 Naming schemes

Data naming is one of the most important technologies in the NDN architecture which may affect the design of the *Interest* forwarding and routing. In sensor networking, the naming system must accurately describe the sensing task thus allowing:

(i) the sink to fully express what information it needs
(ii) the sensors to precisely describe the sensed data.

We recall that our proposed solution consists of three major sequential phases, namely the neighbor discovery, producer discovery, and the data exchange phase. The producer and *Data* exchange phases have a customized naming scheme which will be explained next.

4.1.1 Naming scheme for producer discovery

In the concerned phase, based on the readable and hierarchical design of NDN naming, we propose the following naming scheme:

/SinkId/Location/Producer coordinates/*Data* prefix

Where:

- *SinkId* refers to the prefix to which the sink responds when it receives an *Interest* packet;
- *Location* is the identifier of the geographic area in which the task is performed. Such an attribute can include GPS coordinates or a logical name e.g., in a building, the rooms could have specific unique names such as room0, or area-51;
- *Producer coordinates* refer to the identifier of the geographic coordinates of the producer node; and
- *Data prefix* is the *Data* name to which the producer responds when it receives an *Interest* from the sink

For example when a content provider (P) that produces temperature data for a specefic area wants to announce its existance to a sink (S0), it sends an interest packet with the name

/S0/area1/X/Y/Z/area1/temperature

4.1.2 Naming scheme for data exchange

Based on how the *Data* was obtained, i.e. using the push-based or the pull-based transmission mode, we define different naming schemes:

- *Pull-based mode:* we recall that in this mode, the sensor does not send out collected *Data* until it receives an *Interest*, that we call the *Fetch Interest*, from the sink node. To express this type of *Data* collection, the Fetch *Interest* carries the following naming scheme:

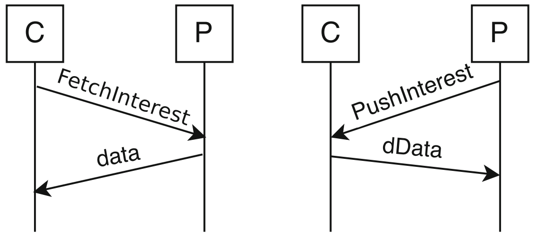
  /Location/Type/Producer coordinates

- *Push-based mode:* in this case the sensor automatically disseminates collected *Data* by pushing it in a special *Interest*, that we call the *Push Interest*, to the sink node. The naming scheme for this *Data* dissemination model is as follows:

  /Location/Type/Sink coordinates/*Data*

where *Type* refers to the name of the sensing task, e.g., temperature, humidity, etc.

The *Push Interest* scheme is derived from the semantics proposed in [15, 17] where arbitrary small *Data* can be included in the *Interest* as a name component since the NDN namespace is potentially unbounded and there are no



(a) Pull-based *Data* **(b)** Push-based *Data*

Fig. 3 Packets flow between the consumer (C) and the producer (P). **a** Pull-based *Data*. **b** Push-based *Data*

restrictions on the number of names components. To ensure reliability, the *consumer* explicitly confirms the reception of the *Push Interest* by sending a dummy *Data* packet (*dData*) used as an acknowledgment.

Figure 3a and b show, respectively, an example of *Push Interest*-dData and *Fetch Interest-Data* exchange between a *producer* P and a *consumer* C in the push-based and the pull-based transmission modes.

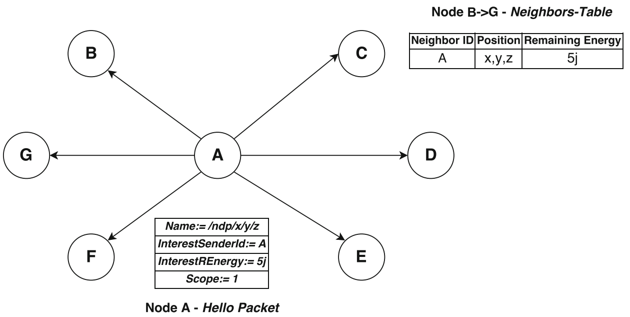
4.2 Neighbor discovery

During the Neighbor discovery phase, each node broadcasts a *HELLO packet* to its direct neighbors. This packet is only used for exchanging location and the remaining energy information between neighbors. This exchange is only done once as a first step to configure the ***Neighbors-Table*** and store information collected by the HELLO packet at each node. Every entry in the *Neighbors-Table* has the following fields: *Neighbor ID*, *Position*, and *Remaining Energy*. The remaining energy value is updated by extracting the remaining energy field inside every received *Interest* packet from a neighbor node. To address this, two new fields, called ***Interest REnergy*** and ***Interest SenderId*** were added to the *Interest* packet to pass the information about the remaining energy at a given node between neighbors.

Figure 4 illustrates the broadcast of the*HELLO Packet*, where Node A wants to announce its existence to its neighbors by sending a Hello Packet that is limited to 1 Hop (specified in the ***Scope*** field) which contains its ID, its position, and its remaining energy. Each node that receives that packet will include node A in its *Neighbors-Table*.

To minimize the power consumption, sensors in GIF do not broadcast a *HELLO packet* periodically. Instead, if a node receives an *Interest* and the sender-id of this packet doesn't correspond to any entry in the *Neighbor-Table* or when a sensor needs to push *Data* to the sink and the *Neighbors-table* is empty, that node initiates the neighbor discovery process.

For example, in Fig. 5 Node B sends an *Interest* Packet that is received by all the nodes in its transmission range, in this example Node A and Node G. Node A discovers that Node B is not listed on its *Neighbors-Table*, as a consequence node A will broadcast a HELLO Packet and every node that receives this packet will verify if the sender is listed on their *Neighbors-Tables*. If the sender ID of the HELLO Packet exists in the *Neighbors-Table* of the receiving node that packet will be ignored, if not that node will also broadcast a Hello Packet to its neighbors.



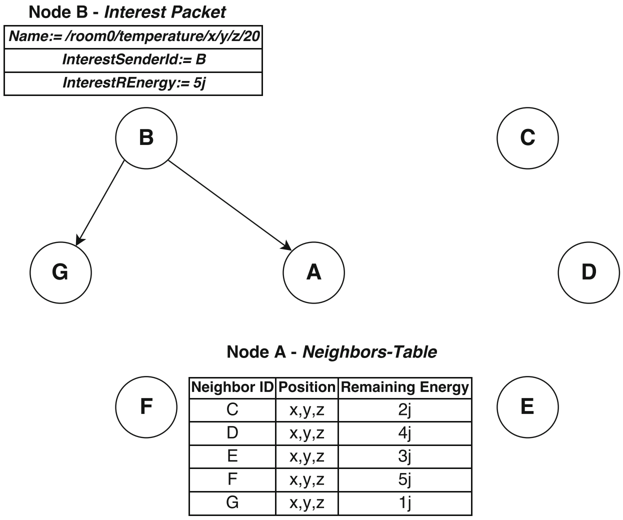
Node B->G - *Neighbors-Table*

Neighbor ID	Position	Remaining Energy
A	x,y,z	5j

Name:= /ndp/x/y/z
InterestSenderId:= A
InterestREnergy:= 5j
Scope:= 1

Node A - *Hello Packet*

Fig. 4 Hello packet broadcast



Node B - *Interest Packet*

Name:= /room0/temperature/x/y/z/20
InterestSenderId:= B
InterestREnergy:= 5j

Node A - *Neighbors-Table*

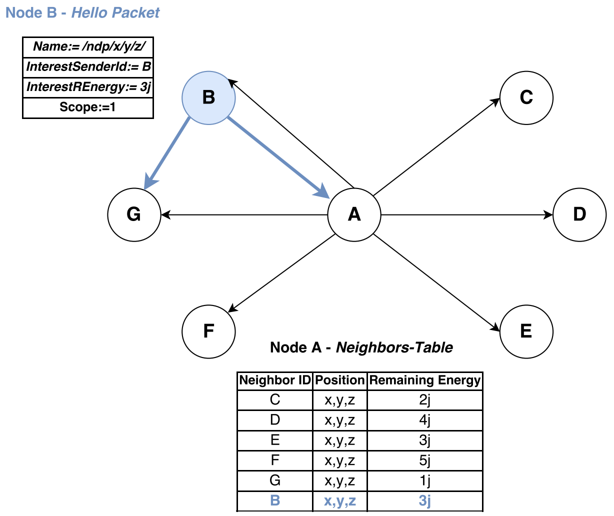
Neighbor ID	Position	Remaining Energy
C	x,y,z	2j
D	x,y,z	4j
E	x,y,z	3j
F	x,y,z	5j
G	x,y,z	1j

Fig. 5 Receiving an interest from an unknown neighbor

This behavior is illustrated on Fig. 6 where Node B receives Node A Hello Packet, it verifies that Node A ID is not listed in its *Neighbors-Table*, then it sends a Hello Packet to announce its existance to Node A.

4.3 Producer discovery

In [7, 8, 10] blind forwarding, called *Flooding*, is used as a first step to discover possible content sources. While



Node B - *Hello Packet*

Name:= /ndp/x/y/z/
InterestSenderId:= B
InterestREnergy:= 3j
Scope:=1

Node A - *Neighbors-Table*

Neighbor ID	Position	Remaining Energy
C	x,y,z	2j
D	x,y,z	4j
E	x,y,z	3j
F	x,y,z	5j
G	x,y,z	1j
B	x,y,z	3j

Fig. 6 Hello packet broadcast from A and reply

“Flooding” is the simplest solution, such technique is resource blind. The *“Flooding”* method does not depend on whether energy resources are scarce or not and it works in the same manner in both two situations. This may have a negative effect on energy consumption especially on large networks. Instead, we used the previously conceived *Push-based Interest* mode (refer to 4.1.2) as a solution for content producers to announce their existence to the sink nodes. When the first phase, i.e. neighbors discovery, is completed and each node has information about its neighbors, the producers send an *Interest* to the sinks announcing their location and the contents name following a greedy forwarding approach which will be discussed in Sect. 4.4. The sinks will store the information sent by the producers in a new table called ***Producers-Table***.

Figure 7 illustrates the producer discovery process where node P wants to announce its *Data* prefix to the Sink node (S0). First, it sends an *Interest* Packet with A as *NextHopID* (the selection of the nexthop will be discussed in details in Sect. 4.4), then when the *Interest* Packet reaches the Sink node the information about the producer will be stored in the ***Producers-Table***.

4.4 Data exchange

In the envisioned static scenario, once the producers have transmitted their information to the sinks, the *Data* query process begins. The main behavior of GIF is illustrated in Fig. 8.

The sink selects a producer from its *Producers-Table* following some criteria, i.e., choosing the closest producer to the sink. Then, before sending an *Interest*, it selects a neighbor that meets certain criteria to forward the packet. When a sensor node receives an *Interest* packet, it will first perform a nexthop check to verify that it was chosen as a forwarder. If a node is not the intended forwarder it will discard the *Interest* packet. This would reduce traffic overhead and collisions by limiting the number of sensors involved in the delivery process. Then, if the sensor node was chosen as a forwarder, it will perform an *Interest* type check. If the received packet is a *Fetch Interest* the forwarding process will continue as described in Sect. 2.1. If the *Interest* packet is a *Push Interest*, the data will be extracted from the packet and it will be added to the content store before continuing the forwarding process. Inspired by the GPSR protocol [18], our approach uses similar techniques to forward *Interest* packets. Each *Interest* is marked by its sender and its destination coordinates as shown in Sect. 4.1.1. As a result, a forwarding node can make a greedy choice by following successively

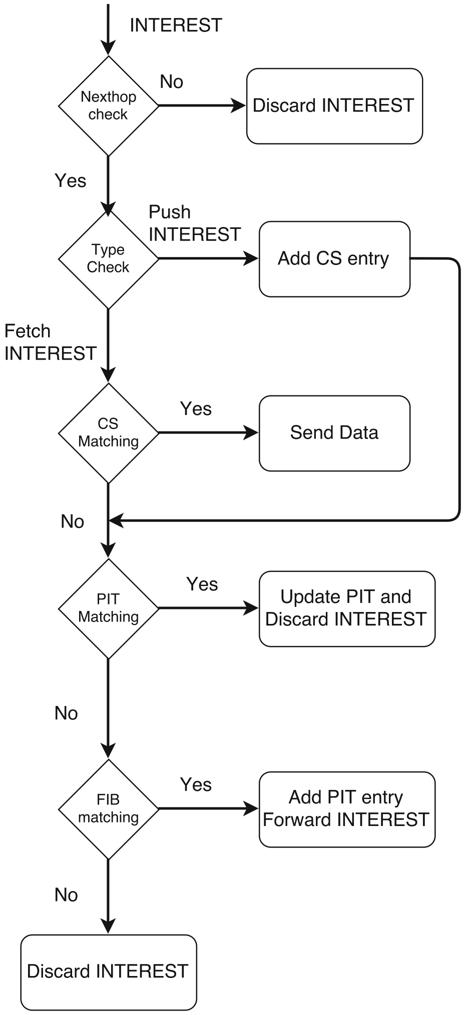
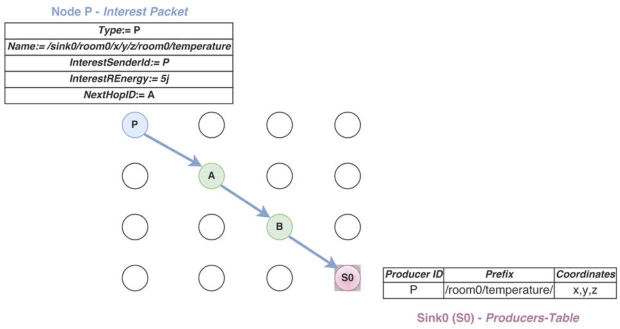


Fig. 8 GIF *Interest* processing



Type:= P
Name:= /sink0/room0/x/y/z/room0/temperature
InterestSenderId:= P
InterestREnergy:= 5j
NextHopID:= A

Producer ID	Prefix	Coordinates
P	/room0/temperature/	x,y,z

Fig. 7 Producer discovery

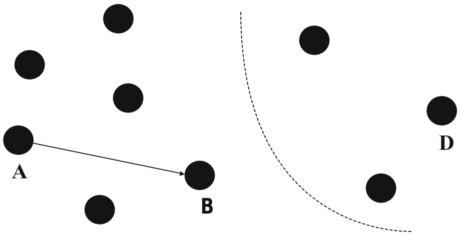


Fig. 9 Greedy forwarding example. B is As closest neighbor to D

closer geographic neighbors until the destination is reached.

Figure 9 shows an example where node A sends an *Interest* to node D. Hence node A forwards the *Interest* to node B which is the closest node to D among all A's neighbors.In order to support more efficient energy forwarding and balance the energy consumption between nodes, a forwarder F_i is chosen based on its distance to the target node and its remaining energy. Let D_{ip} be the distance between a potential forwarder n_i and a target node n_p, D_{tot} the total distance between the consumer and the producer node, E_r (resp. E_{init}) the residual energy (resp. the initial energy) at node n_i. The node with the maximum value F_i is chosen as a forwarder as follows:

$$F_i = \alpha * \left(\frac{E_r}{E_{init}}\right) + (1 - \alpha) * \left(\frac{D_{tot} - D_{ip}}{D_{tot}}\right) \qquad (1)$$

where α is a weight factor to trade off the effects of residual energy and a node distance to the producer in the forwarder selection. We have $0 \leq \alpha \leq 1$. When $\alpha = 0$, the node that is closer to the destination has a higher probability to be chosen to forward *Interest* packets. With the increase of the α value, nodes' remaining energy plays more role in the forwarder selection which helps balance the energy consumption.

In some scenarios, pure greedy geographic forwarding may fail to find a "reasonably optimised" path to the destination, even when one actually exists. To handle this, when a forwarder transmits an *Interest* packet it keeps listening to the channel. If it does not hear the transmission of the *Interest* from the chosen next hop, it concludes that the chosen forwarder failed to deliver the packet to the destination. As a consequence, that node will be removed from the sender *Neighbors-Table*, another forwarder will be chosen and the packet will be retransmitted.

To better explain our proposed solution, we consider the example on Fig. 10. The consumer has the following *Neighbors-Table* illustrated by Table 1.

Table 1 Consumer(C) neighbors-table

ID	Position	Remaining energy
A	x,y	2j
B	x',y'	4j

Before broadcasting the *Interest* packet the *Consumer* (C) computes the F_i value, we choose $E_{init} = 5$, $D_{tot} = 20m$, $\alpha = 0.7$ so that the node that has more residual energy has a higher probability to be chosen to forward *Interest* packets.

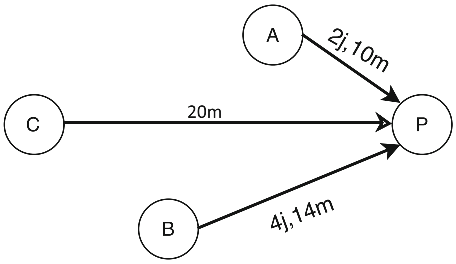


Fig. 10 GIF *Interest* forwarding

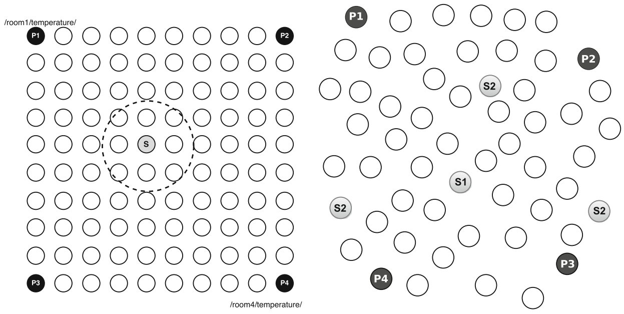
Based on Eq. (1), the computed value of F_A is 0.43 and that of F_B is 0.65 so node B will be chosen as a forwarder. The Consumer then broadcasts an *Interest* which includes the ID of the selected forwarder (This information is included in a new field of the *Interest* packet header) with the name:

/room1/temperature/Producer Coordinates while keep listening to the channel.

When node B receives the *Interest*, it checks if the *Interest* next hop corresponds to its ID. If not the *Interest* will be discarded. Otherwise, B verifies the packet type and forwards it to the next hop. If the Consumer does not overhear the same *Interest*, node B's ID will be removed from the Consumer *Neighbors-table* and the *Interest* will be retransmitted.

5 Simulation model

The aim of the simulation is to evaluate whether a controlled flooding or an aware provider technique is more suitable for wireless sensors networks. To that end, we have developed our proposal based on the NDNSim simulator [19] to evaluate and compare the NDN-based GIF scheme, which uses a provider aware technique, to the directed diffusion NDN scheme (DD-NDN), which is based on a blind forwarding and a controlled flooding approaches to discover possible content providers. The DD-NDN was implemented from scratch based on the description provided in [7].



(a) Lattice Topology **(b)** Random Topology

Fig. 11 Network topologies

In these simulations, the MAC protocol is based on IEEE 802.15.4 and the transmission range of each node is 100m. In the first part of the simulation N sensor nodes are distributed over a 500×500 m^2 lattice topology and in the second part, the sensor nodes were deployed randomly via uniform distribution. Sink nodes are placed in the middle of the topology as shown in Fig. 11a and b. Table 2 lists most of the simulation parameters.

Simulation results are averaged over 20 independent runs and reported with the 95% confidence intervals. We assume that sink nodes send periodically *Interest*s for a specific task. Specifically, we consider a monitoring interval of 60s. The first *Interest* is sent at time t = 120s. Therefore the second *Interest* will be issued 60 seconds later, i.e., at t=180s. The number of monitored tasks is varying from 1 to 16.

In order to evaluate the energy consumption in our simulation we used the following energy model [20]:

$$TX_{i,j} = \epsilon_{elec} * k + \epsilon_{amp} * d_{i,j}^2 \quad (2)$$

$$RX_i = \epsilon_{elec} * k \quad (3)$$

$TX_{i,j}$ denotes the energy cost of transmitting one *Data* packet from node i to node j. RX_i denotes the energy cost for node i to receive a packet. For both Eqs. 2 and 3, $\epsilon_{elec} = 50\mu\mathrm{J/bit}$ is the energy required to run the transmitter or receiver circuitry, $\epsilon_{amp} = 100\,\mathrm{pJ/bit/m^2}$ is the energy required for transmitter amplifier, and k is the number of bits in the packet.

We use five performances metrics described in [21] to evaluate the GIF protocol in term of energy usage, network lifetime, scalability and efficiency. The evaluation metrics are defined as follows:

- **Energy consumption** defined as the total power consumption of all sensor nodes in the network.
- **Network lifetime** defined as the time length until the first sensor node runs out of energy.
- **Number of dropped packets** defined as the total number of dropped *Interest* and *Data* packets.
- **Hop count** defined as the number of intermediate nodes between the Consumer and the Producer through which *Interest*/*Data* packets must pass.
- **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*.

Table 2 Simulation parameters

Parameter	Value
Simulation time (s)	3600
Field size (m^2)	300×300 and 500×500
Wireless interface	IEEE 802.15.4
Number of nodes	64,81,100 and 121
Number of sinks	1 and 4
Communication range (m)	100
Tasks number	1, 4, 8, 12 and 16
Monitoring interval	60s
Interest size (byte)	70
Data size (byte)	120
Initial energy (joule)	5
Energy model	50 μJ/bit [20]

6 Simulations results

Conducted simulations include two parts. In the first part, the DD-NDN and the GIF schemes are compared in terms of **Hop count**, ***Data* retrieval delay** and the protocols reliability to forward *Interest*s and *Data* packets to their destination by measuring the **number of dropped packets**. Also we placed one then four sink node(s) at the middle of the topology as shown in Fig. 11 for lattice and random topologies. In the second part, we compare the energy efficiency of the two *Interest* forwarding schemes, i.e. DD-NDN and GIF, using both one and four sink nodes. We evaluate the effect of increasing the number of *Consumers* and tasks in the network in terms of **energy consumption** and **network lifetime**.

6.1 Comparing hop count, data retrieval delay and the packets drop rate

Figures 12, 13, 14 and 15 illustrate the **Average Hop Count** when 16 tasks are monitored while varying node density from 64 to 121, using 1 and 4 sinks in a lattice and a random topology. The results show that the average hop count achieved by our scheme is significantly lower than the one achieved by DD-NDN which directly affects the retrieval delay. This is due to the fact that, in our

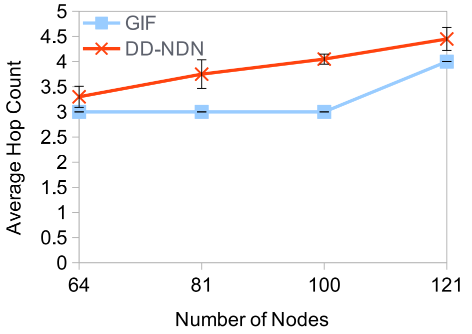


Fig. 12 Average hop count using 1 sink in a lattice topology

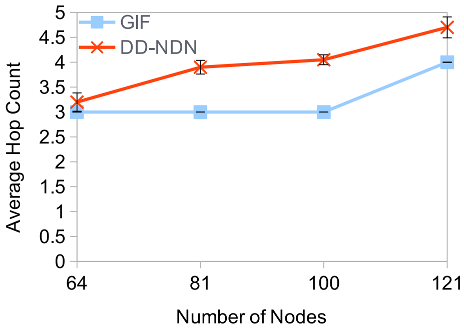


Fig. 13 Average hop count using 4 sinks in a lattice topology

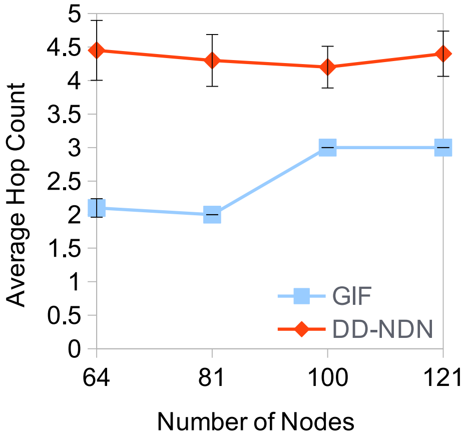


Fig. 14 Average hop count using 1 sink in a random topology

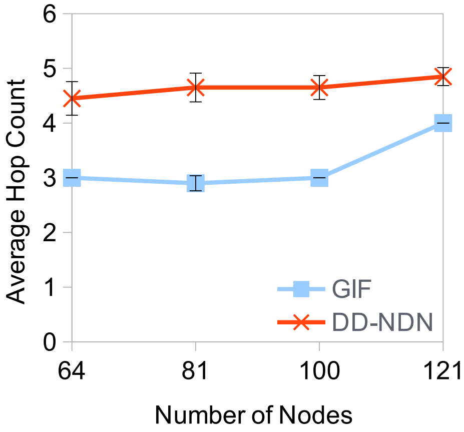


Fig. 15 Average hop count using 4 sinks in a random topology

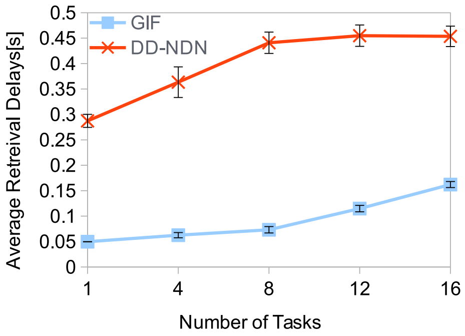


Fig. 16 Average retrieval delay using 1 sink in a lattice topology

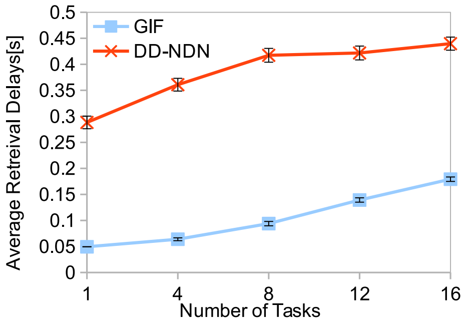


Fig. 17 Average retrieval delay using 4 sinks in a lattice topology

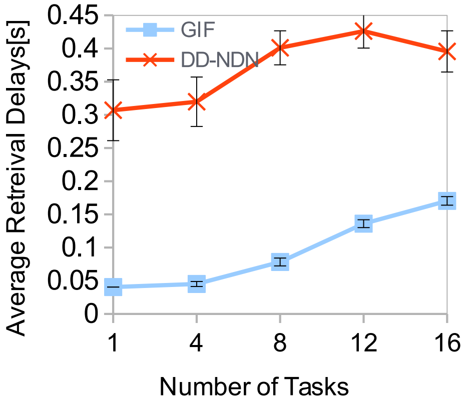


Fig. 18 Average retrieval delay using 1 sink in a random topology

geographic forwarding scheme, *Interest*s are sent from the sink to a producer by choosing the neighbor closer to the destination in each step. Minimizing the hop count, thus, reduces the overall retrieval delay. As expected, the average hop count increases for higher node numbers independently of the used scheme due to the placement positions of the *Producers* and the *Consumer* node. Since the *Producers* are always placed at the extremities of the network and the *Consumers* at the center, the denser the network becomes the further the producer will be placed from the *Consumer* which in turn will increase the average hop count and the average retrieval delay.

In Figs. 16, 17, 18 and 19, we evaluate the effect of different number of monitored tasks on the ***Data*** retrieval delay of the GIF scheme compared with DD-NDN protocol. For these comparisons, the number of nodes is set to 121 nodes. The results show that the retrieval delay achieved by GIF when using one sink node is lower than the one achieved by DD-NDN by 77.58% in lattice topology and by 75.65% in a random topology. Furthermore, when using 4 sinks, the GIF scheme also outperforms DD-NDN by 73.77% and 68.38% in a lattice and a

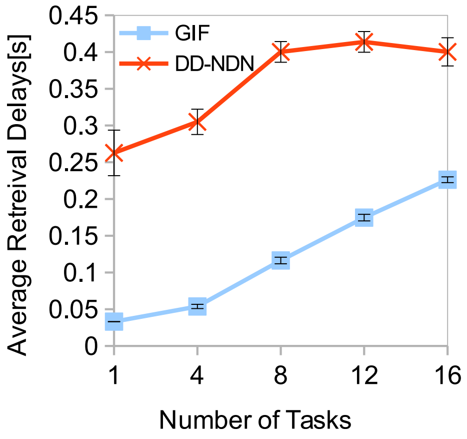


Fig. 19 Average retrieval delay using 4 sinks in a random topology

random topology, respectively. This is due to the following reasons:

- The content consumers in the GIF scheme always choose the nearest provider as its destination;
- The forwarder in our scheme is chosen based on its distance to the provider; and
- The blind controlled flooding in DD-NDN does not always guarantee that the best nodes are selected to forward packets.

Finally in Figs. 20, 21, 22 and 23 we compare both protocols in terms of the **Number of Dropped Packets** since a high number of dropped packets directly affects the energy consumption and the retrieval delay. The results show that in dense and high traffic networks the GIF scheme is more reliable in forwarding packets to their destinations than DD-NDN. For example, for 16 tasks and 4 *Consumers*, GIF doesn't exceed 130 dropped packets while DD-NDN reaches more than 500 dropped packets. This is due to the flooding phase of DD-NDN which causes multiple collisions thus the rate of successful packet transmission will drop.

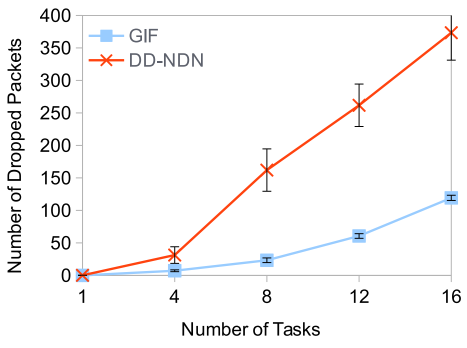


Fig. 20 Number of dropped packets using 1 sink in a lattice topology

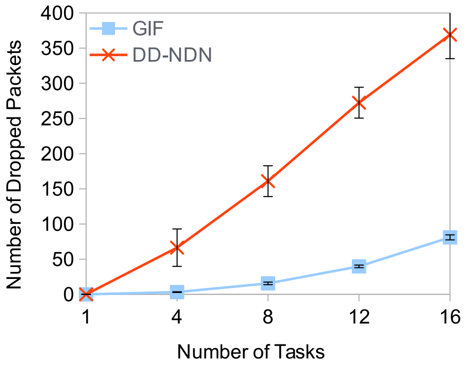


Fig. 21 Number of dropped packets using 4 sinks in a lattice topology

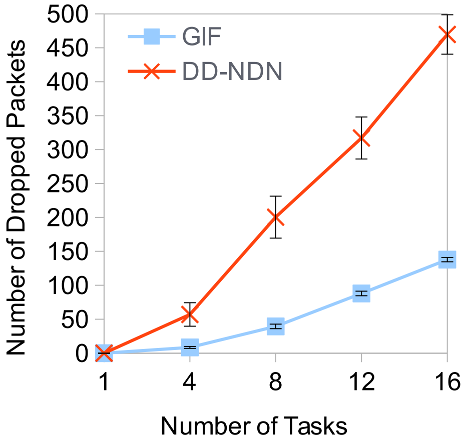


Fig. 22 Number of dropped packets using 1 sink in a random topology

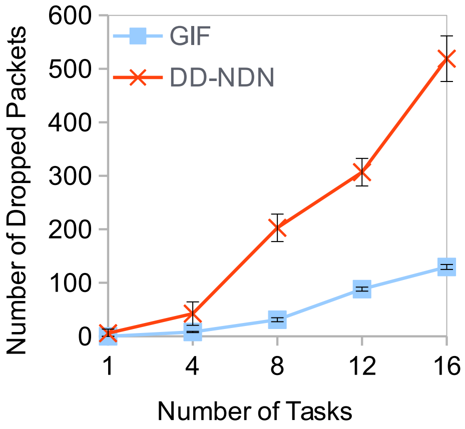


Fig. 23 Number of dropped packets using 4 sinks in a random topology

6.2 Comparing energy consumption and network lifetime

In Figs. 24, 25, 26 and 27, we compare the performance of the proposed GIF with the DD-NDN in terms of **Energy Consumption** and **Network Lifetime** by changing the

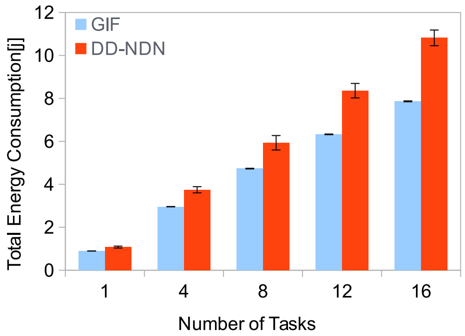


Fig. 24 Total energy consumption using 1 sink in a lattice topology

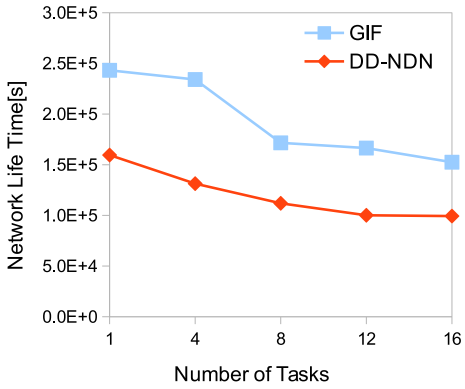


Fig. 25 Network life time using 1 sink in a lattice topology

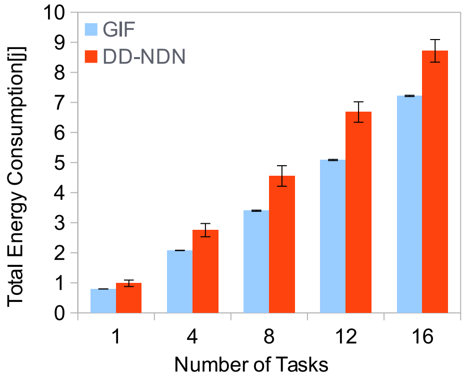


Fig. 26 Total energy consumption using 1 sink in a random topology

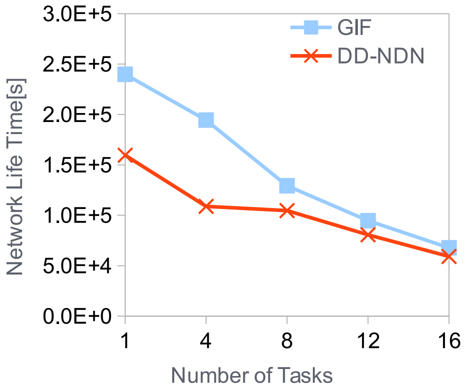


Fig. 27 Network life time using 1 sink in a random topology

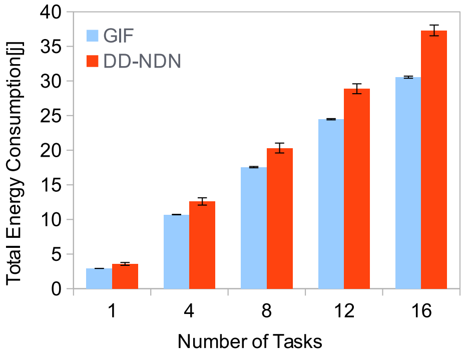


Fig. 28 Total energy consumption using 4 sinks in a lattice topology

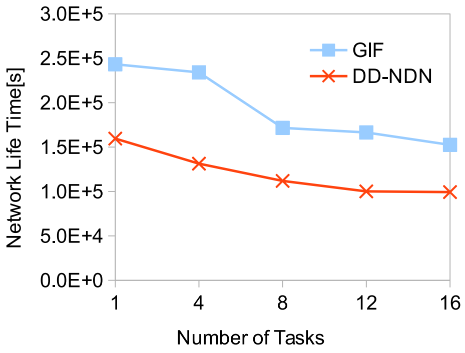


Fig. 29 Network life time using 4 sinks in a lattice topology

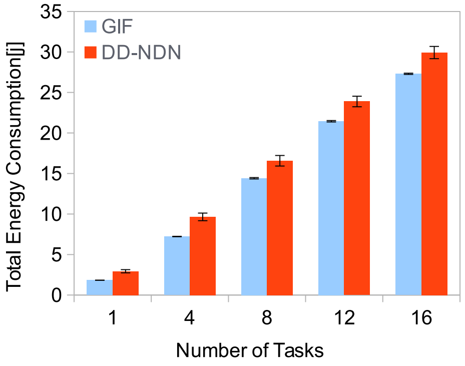


Fig. 30 Total energy consumption using 4 sinks in a random topology

number of tasks from 1 to 16 using only one sink node in both lattice and random topoloy while using 121 nodes. Having more tasks means more *Interest* requests and more energy consumption. In Figs. 28, 29, 30 and 31, we changed the number of sinks from 1 to 4 to evaluate the performance of the two schemes in terms of efficiency in network resources usage as the number of *Consumers* increases. As expected the total energy consumption increases and the network lifetime decreases as the number of tasks and the number of *Consumers* increases. In contrast with DD-NDN, the GIF has better performance in terms of total energy consumption and network lifetime.

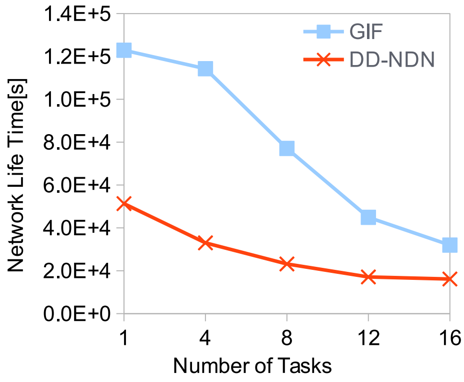


Fig. 31 Network life time using 4 sinks in a random topology

When using one sink node in the network, the bar chart shows that in the GIF scheme the total energy consumption is decreased by an average of 21.95% in a lattice topology and by 22.01% in a random topology, when compared to the DD-NDN. Also GIF scores better results than DD-NDN when comparing the resources usage with an average increase of 24.79% in the network lifetime in a lattice topology and by 37.56% in a random topology. When we increase the number of *Consumers* as shown in Figs. 28, 29, 30 and 31, the GIF scheme still outperforms the DD-NDN protocol in terms of efficiency in network resources usage with a decrease of 16.04% in the total energy usage and an increase of 62.13% in the network lifetime when using a lattice topology and a decrease of 18.98% in the total energy usage and an increase of 44.20% in the network lifetime when using a random topology. This is due to the fact that the GIF scheme does not need flooding to discover new producers. Whereas, in the DD-NDN scheme, the blind flooding phase is started if a selected node does not reply to *Interest* after a fixed timer period. Thus, with the increase of the number of tasks and Consumers, the discovery phase in DD-NDN will be used quite frequently due to traffic overheads and collisions leading to an increase in the packets drop rate. As a consequence, energy consumption will be increased and the network lifetime will be reduced.

6.3 Discussion

The results presented in the previous sections demonstrate that the GIF scheme outperforms the DD-NDN protocol and that it can improve the network performance significantly in terms of resources usage and the data retrieval efficiency, either in a lattice or a random topology. We observed that when the number of tasks is increased and the network traffic is high, the average retrieval delay and the amount of dropped packets obtained by the GIF protocol are lower than the DD-NDN. Similarly, both in small and dense networks, the GIF scheme gives better results in term of the average hop count which directly affects the energy consumption and the retrieval delay of the network. Also, we observed that when comparing the total energy consumption of both scheme, the GIF scheme is slightly better when the network traffic is low. This is due to the neighbors and producers discovery phases of our protocol. But with the increase in the number of tasks and the number of consumers it is clear that the results of the GIF scheme are better than DD-NDN both in the total energy consumption and the network lifetime. In general, our protocol manages to better face the increase in the network density and the network traffic compared to the DD-NDN protocol for which its performances significantly decrease with the increase of the sensor nodes number and the number of tasks.

7 Conclusion

In this paper, we proposed a Geographic Interest Forwarding scheme called GIF for NDN-based IoT. This scheme is based on avoiding flooding techniques since they are resource blind so that the network overhead can be reduced greatly. In addition, several energy efficient mechanisms including broadcast storm avoidance and packet suppression are proposed to save and balance the energy consumption across the network. Analysis and simulations were conducted to evaluate the performance of the proposed NDN-based scheme. The results clearly show that the GIF NDN-based scheme outperforms the directed diffusion NDN-based scheme in term of data retrieval delay. The results also show that using the GIF scheme reduces the total consumed energy since *Interest* packets are never flooded but forwarded through the most suitable neighbor in terms of distance to the destination and sensor remaining energy. Hence, the provider-aware approach proved to be more efficient than the blind forwarding technique since it scales well with the traffic load and the network density thus achieving better results both in terms of usage of network resources and efficiency in the data delivery. However, the results show that the GIF scheme is better suited for dense networks with high traffic load due to the 2 first phases i.e. Producer-discovery anf Neighbor-discovery.

In a future work, we intend to include a security extension to our scheme and to extend our performance study to more complex scenarios, at a larger scale while considering the nodes mobility and the sleep mode of 802.15.4.

References

1. Samani, A., Ghenniwa, HH., Wahaishi, A.: Privacy in internet of things: a model and protection framework. Procedia Comput. Sci. 52(Complete):606–613 (2015)
2. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless sensor networks: a survey. Comput. Netw. **38**(4), 393–422 (2002)
3. Palattella, M.R., Accettura, N., Vilajosana, X., Watteyne, T., Grieco, L.A., Boggia, G., Dohler, M.: Standardized protocol stack for the internet of (important) things. IEEE Commun. Surv. Tutor. **15**(3), 1389–1406 (2013)
4. Zhang, L., Afanasyev, A., Burke, J., Jacobson, V., Claffy, K., Crowley, P., Papadopoulos, C., Wang, L., Zhang, B.: Named data networking. SIGCOMM Comput. Commun. Rev. **44**(3), 66–73 (2014)
5. Jacobson, V., Smetters, DK., Thornton, JD., Plass, MF., Briggs, NH., Braynard, RL.: Networking named content. In: Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies, ACM, New York, NY, USA, CoNEXT '09, pp 1–12 (2009)
6. Singh, S.K., Singh, D.K., Singh, M.P.: Routing protocols in wireless sensor networks a survey. Int. J. Comput. Sci. Eng. Surv. **1**(2), 63–83 (2010)
7. Amadeo, M., Campolo, C., Molinaro, A., Mitton, N.: Named data networking: a natural design for data collection in wireless sensor networks. In: Wireless Days (WD), 2013 IFIP, pp 1–6 (2013)
8. Amadeo, M., Campolo, C., Molinaro, A.: Forwarding strategies in named data wireless ad hoc networks: design and evaluation. J. Netw. Comput. Appl. **50**, 148–158 (2015)
9. Amadeo, M., Campolo, C., Molinaro, A.: Multi-source data retrieval in iot via named data networking. In: Proceedings of the 1st International Conference on Information-centric Networking, ACM, New York, NY, USA, ICN '14, pp 67–76 (2014)
10. Gao, S., Zhang, H., Zhang, B.: Energy efficient interest forwarding in ndn-based wireless sensor networks. Mob. Inf. Syst. **2016**, 15 (2016)
11. Burke, J., Gasti, P., Nathan, N., Tsudik, G.: Securing instrumented environments over content-centric networking: the case of lighting control and ndn. In: 2013 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp 394–398 (2013)
12. Wang, C., Li, J., Ye, F., Yang, Y.: Netwrap: an ndn based real-time wireless recharging framework for wireless sensor networks. IEEE Trans. Mob. Comput. **13**(6), 1283–1297 (2014)
13. Gusev, P., Burke, J.: Ndn-rtc: real-time videoconferencing over named data networking. In: Proceedings of the 2Nd ACM Conference on Information-Centric Networking, ACM, New York, NY, USA, ACM-ICN '15, pp 117–126 (2015)
14. Intanagonwiwat, C., Govindan, R., Estrin, D., Heidemann, J., Silva, F.: Directed diffusion for wireless sensor networking. IEEE/ACM Trans. Netw. **11**(1), 2–16 (2003)
15. Amadeo, M., Campolo, C., Molinaro, A.: Enhancing content-centric networking for vehicular environments. Comput. Netw. **57**(16), 3222–3234 (2013a)
16. Amadeo, M., Molinaro, A., Ruggeri, G.: E-chanet: routing, forwarding and transport in information-centric multihop wireless networks. Comput. Commun. **36**(7), 792–803 (2013b)
17. Amadeo, M., Campolo, C., Molinaro, A.: Internet of things via named data networking: the support of push traffic. In: 2014 International Conference and Workshop on the Network of the Future (NOF), pp 1–5 (2014)
18. Karp, B., Kung, HT.: Gpsr: Greedy perimeter stateless routing for wireless networks. In: Proceedings of the 6th Annual International Conference on Mobile Computing and Networking, ACM, New York, NY, USA, MobiCom '00, pp 243–254 (2000)
19. Mastorakis, S., Afanasyev, A., Zhang, L.: On the evolution of ndnSIM: an open-source simulator for NDN experimentation. ACM Comput. Commun. Rev. **47**, 19–33 (2017)
20. Ko, M., Korpeoglu, I.: Controlled sink mobility algorithms for wireless sensor networks. Int. J. Distrib. Sens. Netw. **10**(4), 167,508 (2014)
21. Mamun, Q.: A qualitative comparison of different logical topologies for wireless sensor networks. Sensors **12**(11), 14,887–14,913 (2012)

Ahmed Aboud is a Ph.D. Student in Computer Science and Networking at the National Engeneering School of Sfax (ENIS) in Tunisia. He has a Bachelors degree in Computer Science and a Masters degree in Computer Networks from the Faculty of Sciences of Gabes, Tunisia. His research is focused on designing efficient and energy aware routing protocols for the internet of things (IoT).

Haifa Touati received her Ph.D. degree in Computer Science at the National School of Computer Science, University of Manouba, Tunisia in 2011. Previously, she earned her engineering and master degrees in networking at the National School of Computer Science, Tunisia. Since 2004, she holds the position of lecturer then assistant professor at the Faculty of Science de Gabes in Tunisia. She was the supervisor of the Master's degree program in Systems and Networks Security. Her research focuses on issues related to named data architecture and wireless networking (ad hoc, sensor, VANET): efficient routing protocols design, networks modeling and performances, congestion control and energy conservation.

Brahim Hnich is a Professor of Computer Science (CS) at the university of Monastir, Tunisia. He holds a B.Sc. in Computer Engineering from Bilkent University (1997, Turkey), a Ph.D. in CS from Uppsala University (2003, Sweden), as well as a HDR in CS from Montepellier II University (2008, France). He coauthored more than 100 journal/conference papers. His main area of research is constraint programming.