



LAFS: a learning-based adaptive forwarding strategy for NDN-based IoT networks

Adel Djama¹ · Badis Djamaa¹ · Mustapha Reda Senouci¹ · Nabil Khemache¹

Received: 2 September 2020 / Accepted: 21 April 2021 / Published online: 14 July 2021
© Institut Mines-Télécom and Springer Nature Switzerland AG 2021

Abstract

Named Data Networking (NDN) is a data-driven networking model that proposes to fetch data using names instead of source addresses. This new architecture is considered attractive for the Internet of Things (IoT) due to its salient features, such as naming, caching, and stateful forwarding, which allow it to support the major requirements of IoT environments natively. Nevertheless, some NDN mechanisms, such as forwarding, need to be optimized to accommodate the constraints of IoT devices and networks. This paper presents LAFS, a Learning-based Adaptive Forwarding Strategy for NDN-based IoT networks. LAFS enhances network performances while alleviating the use of its resources. The proposed strategy is based on a learning process that provides the necessary knowledge allowing network nodes to collaborate smartly and offer a lightweight and adaptive forwarding scheme, best suited for IoT environments. LAFS is implemented in ndnSIM and compared with state-of-the-art NDN forwarding schemes. As the obtained results demonstrate, LAFS outperforms the benchmarked solutions in terms of content retrieval time, request satisfactory rate, and energy consumption.

Keywords Internet of things · Named data networking · Stateful forwarding · Low-power wireless networks

1 Introduction

The massive proliferation of low-cost smart objects in our life, combined with their inevitable connection to the global Internet, for remote monitoring and control purposes, in particular, has led to the well-known Internet of Things (IoT), where the majority of application fields, such as smart healthcare, smart agriculture, and smart monitoring, necessitate the deployment of wireless ad hoc-based networks. Interconnecting the resource-constrained objects in this new digital sphere is currently supported largely by

IP-based technologies, which rely on the enhancement of the original TCP/IP stack to fit IoT basic needs. Nevertheless, these adaptation efforts have incurred extra complexity and overhead on the network resources [1].

Recent researches explore the capabilities of the Information-Centric Networking (ICN) paradigm in handling IoT requirements, which proposes a new way to do networking by fetching Data by names instead of their host IP addresses. This new networking technology would provide native support for the majority of the existing IoT applications where Data is considered as the primary focus.

Named Data Networking (NDN) [2] is considered as the most prominent instantiation of ICN. NDN's inherent features, such as caching, naming, packet-level security, and stateful forwarding plane, make it extremely suitable for wireless ad hoc environments, such as Mobile Ad hoc Networks (MANETs), Vehicular Ad hoc Networks (VANETs), or Wireless Sensor Networks (WSNs) that constitute an integral part of the IoT ecosystem. In such networks, the access concurrency on the wireless medium that causes packets collisions and losses, added to the frequent topology changes due to, among others, nodes mobility pose a problem to the forwarding strategies, which have to invoke additional mechanisms to cope with the dynamic and unreliable nature of these environments.

✉ Adel Djama
a.djamaa@esi.dz

Badis Djamaa
badis.djamaa@emp.mdn.dz

Mustapha Reda Senouci
mustaphareda.senouci@emp.mdn.dz

Nabil Khemache
khemache.nabil@gmail.com

¹ Distributed and Complex Systems Lab., Ecole Militaire Polytechnique, Algiers, Algeria

Indeed, content retrieval in conventional IP-based ad hoc architectures is mainly based on routing strategy, which is considered the smart operation, while the forwarding action is less important since its unique role is to concretize the routing decision by transmitting the packet hop by hop until reaching the final destination. In fact, IP-based routing solutions have been widely explored in this design space [3], i.e., ad hoc environment, where the research focus is based on the support of its various inherent constraints, such as resource limitations (energy and processing), and the dynamic and unpredictable location of the nodes. In this scope, several solutions have been proposed, such as [4] to overcome dynamic holes through multipath forwarding in Wireless Multimedia Sensor Networks, or [5] for the support of the nodes' energy depletion and fault tolerance in WSNs.

In addition to the routing mechanism, NDN employs a stateful forwarding plane, where each node keeps track of the incoming Interests to respond to them later, positively or negatively, depending on whether the matching Data is obtained. That is why the forwarding operation in NDN is considered adaptive and smart in contrast to the IP-based one. Moreover, thanks to its salient characteristics, especially naming and caching, which are implemented directly at the network layer, multicast/anycast forwarding are natively supported, hence offering robust support to cope with dynamic network topology changes of the ad hoc environments in general and IoT in particular.

Nevertheless, with the existence of one external communication face, NDN forwarding operation faces a new range of challenges in such ad hoc scenarios, especially the broadcast storm problem [6]. This phenomenon is not properly handled by the NDN native design and may cause significant network performance degradation. Consequently, the research community has devoted an effort focused on forwarding solutions to overcome this new challenge, but this is still at the beginning stage [7].

The main contributions of the present work can be summarized as follows:

- We highlight the NDN-based forwarding peculiarities for wireless ad hoc networks;
- We conduct a comparative analytical study of the existing forwarding solutions in NDN-based wireless ad hoc networks;
- We devise a new Learning-based Adaptive Forwarding Strategy (LAFS) tailored for NDN-based wireless IoT networks;
- We propose a smart healthcare system for COVID-19 confinement area monitoring, as a relevant use case; and

- We demonstrate, by simulations, that our forwarding solution enhances the performance of the network while preserving its resources in comparison to other competitive approaches.

The proposed protocol (LAFS) employs two novel techniques; a learning mechanism and an adaptive forwarding function allowing together a smart collaboration between IoT nodes while respecting their constrained resources.

The rest of this paper is organized as follows. Section 2 introduces Named Data Networking and its peculiarities in wireless ad hoc networks. Section 3 follows this by presenting a comprehensive comparative study on the existing NDN-based forwarding techniques for wireless ad hoc networks. In Section 4, our new forwarding strategy, adapted for NDN-based wireless IoT networks, and its main phases, operations, and parameters are presented. The performance evaluation of our proposal is compared to other competitive solutions and presented in Section 5. In Section 6, the conclusion, the main obtained results, and insights on future work are summarized.

2 Named data networking in wireless ad hoc networks

This section is devoted to present the basic architecture and main challenges of the NDN-based wireless ad hoc environment.

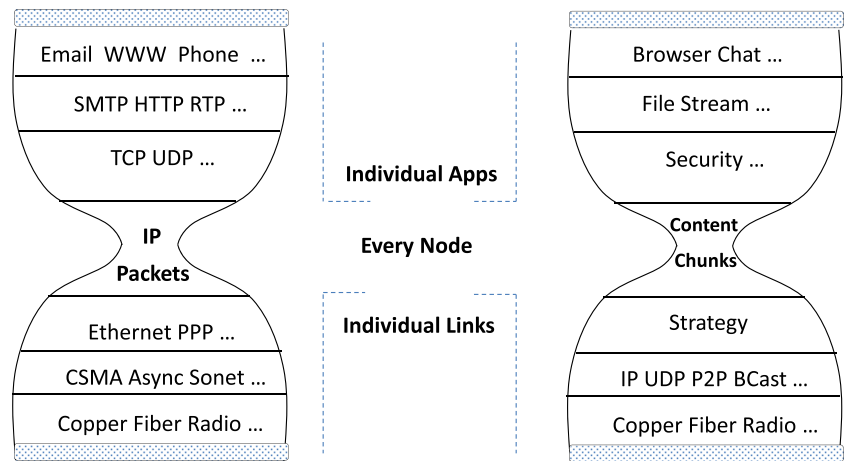
2.1 NDN-based wireless ad hoc architecture

NDN inherits the IP architecture hourglass shape, but substitutes the end-to-end data delivery model with a receiver-driven data retrieval model at the thin waist level, as illustrated in Fig. 1. This new design leads to the shift in the communication paradigm from location-centric (*where*) to data-centric (*what*).

NDN architecture suggests, through the decoupling of the sender from the receiver, to retrieve Data directly on the network layer of the communication hosts by relying on application Data names instead of source addresses.

The services in NDN are based on retrieving Data identified by a name. Therefore, only two types of packets are used: Interest and Data (Fig. 2). A consumer fetches Data by sending an Interest, which disseminates a name that identifies the targeted Data. A Data packet is transmitted only in response to an Interest and consumes that Interest. The Data packet, which follows the reverse path taken by

Fig. 1 NDN vs. IP stack



the Interest to get back to the consumer, can be generated by any node having the desired Data, like the producer or in-network storage: persistent storage (repository) or temporary storage (caching router). In fact, in NDN, a router can cache the Data packets in its content store and uses them to satisfy further requests.

In a wireless ad hoc architecture, where all NDN nodes use and share one medium access technology, e.g. IEEE 802.15.4, in addition to the node’s internal face between network and application layers, there is only one external face that is used by the node to exchange all types of packets with its neighbors, especially Interest and Data ones.

Figure 3 depicts the structure of a NDN wireless ad hoc node, projected from the network layer, where three data structures are used: CS (Content Store), PIT (Pending Interest Table), and FIB (Forwarding Information Base). The CS is employed to cache temporary Data packets, which reduces the communication time in the network. The PIT is used to save the incoming faces of pending Interests and to respond to them later once the matched Data is obtained; otherwise, a negative acknowledgment is returned. The last data structure, i.e., the FIB, that is populated by a routing protocol, contains Data name prefixes and the corresponding output faces toward potential Data provider(s). Besides, these three data structures are simultaneously exploited, by a forwarding strategy engine, namely Ndn Forwarder Daemon (NFD) [8], to make the forwarding decisions about incoming Interest or Data packets.

In the following section, we briefly describe the main challenges facing NDN in wireless ad hoc networks, with a focus on the forwarding plane.

2.2 Main challenges in NDN-based wireless ad hoc environment

In NDN-based wireless ad hoc environment, although the NDN node basic structure remains unchanged, compared to infrastructure architecture, some specific peculiarities impact the functioning of its forwarding engine NFD due to the wireless nature of the communication medium, which incurs additional challenges to deal with.

In fact, in these networks, content retrieval is mainly based on forwarding operation, which consists of transmitting the packet (Data or Interest) from the sender, hop by hop, until reaching the destination. Furthermore, since node addresses (or identifiers) are not integrated as a packet field in the native NDN architecture, it relies on the broadcast operation through the unique external face, for both Interest and Data transmission phases, which is considered as the fundamental communication mode in ad hoc networks [10].

Indeed, in the Interest transmission phase, the consumer diffuses an Interest packet to its direct neighbors, which similarly repeat the same operation, and so on until reaching a Data provider. This repeated diffusion action, of the Interest packet leads to the well-known broadcast storm problem in wireless ad hoc networks.

Fig. 2 NDN packets

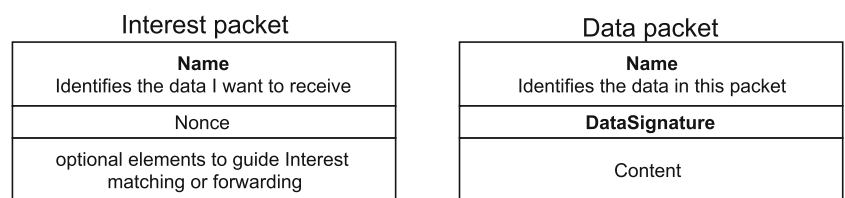
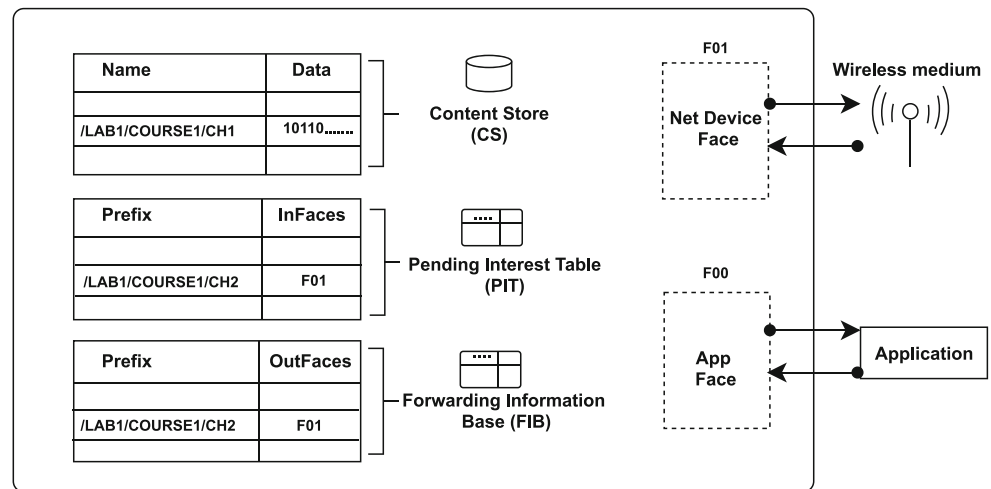


Fig. 3 NDN wireless ad hoc node structure, adapted from [9]



In the Data transmission phase, although the source node broadcasts the Data packet to all its neighbors, only the requesting nodes, whose PIT table contains the corresponding prefix name, will treat the Data packet and forward it further until it arrives at the initial consumer. When a receiver node does not find a corresponding PIT entry for the incoming Data packet, the latter will be automatically dropped. Thus, it can be observed that NDN natural design permits attenuating significantly the effect of the broadcast storm in the Data transmission phase.

Moreover, it is worth mentioning that NFD, in its latest versions, has been updated to support ad hoc link-based communication in addition to wired-based one, by bringing a set of specific conditions and/or operations while managing events related to Data and Interest packets.

For instance, ad hoc mode allows broadcasting Interest or Data packets on the same incoming face since there is one external communication face. However, this case is prohibited in point-to-point communication mode (wired) where each communication link has its corresponding face. Also, sending NACK packets operation is disabled or bypassed in ad hoc mode, so even a neighbor node with no valid next hop (outgoing face in the FIB table) does not send a NACK packet on the incoming face of that Interest, since the targeted Data could be brought to the requester by another neighbor node sharing the same ad hoc face. Lastly, for ad hoc communication mode, when a node receives for the first time an Interest packet, it keeps it in the PIT table, and any further reception of the same Interest is considered a loop, and the packet is systematically dropped; whereas, in wired point to point communication mode, duplicate Interest from the same incoming face is not considered a loop.

To sum up, we can say that the simple and flexible design of NDN, thanks to its key strengths, namely naming, caching, inherent multicast/anycast support, and stateful

forwarding, allows it to handle efficiently the unreliable and dynamic aspects of the wireless ad hoc networks in general and the IoT in particular.

After highlighting NDN core principles and challenges in the context of wireless ad hoc networks, from a functional point of view, in the following section, we will focus on the NDN-based forwarding strategies proposed in this area.

3 Related work

In this section, we present a comprehensive review of the existing NDN forwarding strategies designed for wireless ad hoc environments. Besides, since we are interested in NDN-based IoT architecture as a research area, our analysis of the surveyed literature will be mainly oriented to appreciate their suitability for this constrained environment, especially in terms of respect to the NDN fundamental machinery as well as the lightweight design reflected basically by the amount of resources required by the nodes in the network (memory, processing, energy, and bandwidth).

The work presented in [11] proposes an Interest forwarding solution, based on randomized scheduling timers, to minimize packet collisions while using geo-location of nodes to perform distance-based Data dissemination. As a first observation, the use of geo-coordinates reflects the position of the nodes, which is opposed to the NDN principle, where the networking is supposed to be centered on the Data rather than its location. Moreover, geo-coordinates of nodes, especially in a three-dimension environment, necessitate a great memory space to be stored in the node, in addition to an extra geo-location module (e.g., GPS) for the determination of these coordinates, which is not suited for constrained IoT deployment. Finally, the randomized scheduling timer, used in this solution does not guarantee systematically an optimized path to the destination.

A forwarding algorithm for wireless ad hoc NDN networks is proposed in [12] while relying on beacon messages to decide whether to forward an Interest or not. These exchanged beacons include the identifier of the sender and a list of all its valid neighbors carried in a bloom filter field. The latter is compared with the one of the receiver, and the Interest will be forwarded to nodes that are not included in the incoming bloom filter. In this solution, an extra packet type is used comparatively to NDN native architecture, namely beacon messages. This induces a waste of energy of the nodes and an additional overhead on the network bandwidth, which is not adapted for IoT applications. Also, in the case of a dense network, the size of the bloom filter could exceed the MTU adopted by the most common IoT communication protocols, such as the IEEE 802.15.4, leading to packet fragmentation.

In [13], an enhancement of the NDN forwarding scheme in WSNs is proposed by including the principles inspired by the Directed Diffusion protocol [14]. To this end, the legacy NDN Data packet has been overloaded to carry the identifier of the sender, which is stored in a new data structure called the Next Hop Table (NHT). This latter is used by the nodes to guide incoming Data retrieval queries. Besides, the same authors propose in [15] a content-centric architecture to cope with multihop wireless networks (E-CHANET). The forwarding operation is based on a new distance table stored in the node and extracted from the exchanged packets (Interest and Data), which includes the provider ID and the distance to the requester. These two last described strategies involve the use of new table structures in forwarding decisions while preserving the legacy NDN packets. These new tables are not necessary from our point of view and could be integrated into the native NDN data structures, such as the FIB table, which permits to alleviate the occupied memory of the constrained nodes.

Moreover, the authors in [16] propose a Neighborhood-Aware Interest Forwarding (NAIf) protocol for MANETs. The eligibility of a forwarder node is calculated based on its Data retrieval rate for the given prefix name and its distance to the requester. The aim is to decrease the Interest flooding and alleviate the bandwidth of the network by adjusting the sending rate of Interest packets. Nevertheless, no mechanism has been used to prevent the broadcast storm of the unknown Interest packets, which could lead to serious performance degradation, especially in the case of constrained IoT networks. On the other hand, in [17], a direction-selective forwarding strategy for content retrieval is proposed in the context of mobile cloud computing, where a forwarder node relies on the geographical coordinates of its neighbors and additional packets (ACK and CMD) to choose the relay nodes according to the four quadrants of its transmission range. As we can see, similar to our analysis of paper [11], using geo-coordinates, in addition to the fact

that it invokes host-centric communication rather than a data-centric one, is not adequate in IoT deployments.

In [18], the authors designed a forwarding mechanism called Reactive Optimistic Name-based Routing (RONR), which reduces the number of radio transmissions and receptions in the IoT environment. Only initial Data request flooding is needed, while the subsequent Interests, carrying the same prefix, follow the reverse path, taken by the initial Data response that is populated in the FIB table, hop by hop until reaching the producer. It can be observed that this proposed strategy has been designed with respect to almost all NDN original components (packets and data structures), which is convenient with constrained IoT basic concerns. However, no support of mobility or failure of nodes has been provided in this work, which count also among the main requirements of IoT applications.

The authors in [19] propose a greedy Geographic Interest Forwarding (GIf) scheme for NDN-based IoT. To this end, a Neighbors Discovery phase is performed via HELLO messages, including the ID of the sender and its coordinates, and a Producers Discovery phase is initiated by content producers to announce their existence to the consumers before performing Data queries. It can be observed that the proposed forwarding scheme uses the geo-coordinates of nodes in addition to an extra packet type (HELLO message), which could be costly in resource consumption, especially in the case of mobile resource-constrained IoT network.

Besides, a Dual-Mode Interest Forwarding scheme (DMIf) for NDN-based WSNs is proposed in [20], where an alternation between flooding mode and directive mode is adopted by nodes depending on the FIB lookup for the incoming Interests. To control the Interest flooding scope, a TTL field is added to the Interest packet, which is automatically adjusted according to the network requirements. Also, a deferred timer for the Interest and Data forwarding processes is used to overcome broadcast storms in wireless networks. As a remark to this research work, in a NDN-based wireless ad hoc scenario, using a deferred timer in the data forwarding phase has no impact on the broadcast storm, because this latter problem is natively handled by the basic NDN architecture, thanks to its stateful forwarding plane. Consequently, the proposed solution is not efficient in packet retrieval time and, thus, not suited in real-time like applications.

In another research work [21], a hybrid forwarding mechanism is proposed for content delivery in wireless ICN, where an Ad hoc Dynamic Unicast (ADU) method was designed on top of the MAC layer to allow dynamic switching between unicast and broadcast communication modes based on MAC notifications. For this, ADU uses MAC addresses that are carried in Data packets and stored as an entry in the FIB table, which will serve as the next hop for subsequent queries with the same prefix name. This

proposed forwarding solution, it can be noted, exploits the MAC addresses for Interest packets transmission, which is logically closer to host-centric than the data-centric paradigm.

The authors in [22] propose a NDN-based forwarding strategy for low-end IoT, which is based on a reinforcement learning technique to adjust the waiting time of a forwarder node before broadcasting the Interest packet. By doing so, no extra packets or data structures are needed by the protocol, except a cost field that is carried in both Interest and Data packets, and which reflects the distance of a node to the provider and its eligibility to forward an Interest packet. This proposed solution involves a reinforcement learning technique at every forwarding node, which requires significant processing capabilities to achieve this machine learning process continuously (at every exchanged packet). Thus, this approach is less recommended for implementation in low computational IoT devices.

Lastly, paper [23] introduces a Location-Based Deferred Broadcast (LBDB) scheme for ad hoc NDN networks. A collision avoidance timer is used in the Interest dissemination phase, which reflects the forwarding priority of a node and is mainly based on the location information of the forwarding node and the data sources. The goal is to reduce data delivery delay by alleviating transmission overhead. As this proposed forwarding scheme is based on the location of the nodes, especially the data sources, we consider that it somehow violates the NDN principle, namely the data-centric communication paradigm. Besides, the use of geo-coordinates is not adequate for resource-constrained IoT networks.

Table 1 summarizes the reviewed NDN-based forwarding solutions in wireless ad hoc networks and their main peculiarities projected according to six comparison dimensions (columns). The first column “*Extra packets/fields or data structures*” reflects the use of any additional packets or structures in comparison to the original NDN. “*Deferred broadcasting*” column depicts whether a timer is used or not by the proposed strategies during the packet dissemination process and for which purpose. Besides, the “*Lightweight design*” column describes the lightness degree of the proposed forwarding mechanisms from a functional viewpoint and their suitability for IoT environments. The “*Targeted area*” criterion represents the targeted application area of the proposed research, whereas the “*Link Layer*” dimension depicts the employed wireless communication protocol (layer 2) by the proposed solutions. Finally, the last column, “*Evaluation platform*”, shows the used evaluation tools, for example, ndn SIMulator (ndnSIM) [24], the official simulator of the NDN project, CCN-Lite [25], and RIOT [26], which together enable CCN-based IoT deployment in addition to other evaluation tools, such as

Qualnet [27], CORE [28], NS2 [29], OMNeT++ [30], and Arduino [31].

To sum up, what we can learn from our literature review is that all the proposed solutions do not respect properly NDN native principles, where extra fields are added to the original packets (Interest and Data), such as identifiers of nodes, or extra data structures are used to keep information about the network activity. Moreover, deferred broadcasting timers are usually used to avoid broadcast storm problem and improve packet delivery performance in wireless ad hoc networks. However, these modifications, if not well handled and optimized, could incur extra complexity and overhead (heaviness) to the resource-constrained nodes.

Based on these lessons, we will present in the next section, our forwarding solution adapted for NDN-based IoT networks. Our main contribution compared to the reviewed research resides in its lightweight design, through bringing minor changes to the original NDN structures, while keeping its native machinery, in addition to a learning-based forwarding mechanism that allows the dynamical handling of the main constraints of the IoT environment without depleting its resources.

4 Design and description of our proposal

In this section, we provide an in-depth description of our main contribution, which is of a new forwarding strategy for NDN-based IoT networks, baptized “*LAFS: Learning-based Adaptive Forwarding Strategy*”.

4.1 Overview

We have learned from our study presented in Section 3 that a minor number of researches have been devoted to NDN-based IoT networks. Existing forwarding approaches basically use two technique, deferred broadcasting and/or modification of NDN primitives, to overcome constraints imposed by wireless ad hoc communication links while incurring different over costs on the network. Thus, our objective is to conceive a lightweight forwarding technique adapted for low-end NDN-based wireless ad hoc IoT, which gives the best compromise (trade-off) in terms of network performance metrics, namely energy consumption, overhead, success rate, and content retrieval time.

We strive in our conception to meet the lightweight design of the IoT environment by adopting the following guidelines: (i) No use of extra packets, except those used by NDN native design (Interest and Data packets), to prevent network traffic congestion (overload) and (ii) Minimize the use of extra fields in the original NDN structures (PIT and FIB) and packets (Data and Interest) to meet the

Table 1 NDN-based forwarding solutions in wireless ad hoc networks and main characteristics

Ref.	Extra packets/fields or data structures	Deferred broadcasting	Lightweight design	Targeted area	Link layer	Evaluation platform
Wang et al. [11]	Geo-location of nodes	Collision-avoidance timer	Medium	V2V	IEEE 802.11a	ndnSIM
Angius et al. [12]	Beacon message (ID of sender + bloom filter)	–	Low	WSNs	–	Analytical
Amadeo et al. [13]	ID of sender in Data packet + NHT table	Anti-collision timer	Medium	WSNs	IEEE 802.11g	ndnSIM
Amadeo et al. [15]	Provider ID and Hop Count in Interest and Data packets + Distance table	Packet redundancy control timer	Medium	WSNs	IEEE 802.11g	ndnSIM
Yu et al. [16]	Distance to the consumer (hop count)	–	Medium	MANETs	IEEE 802.11b	Qualnet/ CORE
Lu et al. [17]	Geographical coordinates of nodes + additional packets (ACK and CMD)	Back-off timer before sending ACK	Low	MANETs	IEEE 802.11b	NS2
Baccelli et al. [18]	Temporary entry in FIB table to auto-direct subsequent queries by unicast	–	High	IoT	IEEE 802.15.4	CCNlite/ RIOT
About et al. [19]	HELLO packet (ID of sender, geographic coordinates) + Remaining energy in Interest packet + Neighbors table	–	Low	IoT	IEEE 802.15.4	ndnSIM
Gao et al. [20]	TTL into Interest packet + additional FIB entries (Forwarder ID, Hop, Energy, Timer)	A deferred timer for Interest and Data forwarding against broadcast storm	Medium	WSNs	IEEE 802.15.4	ndnSIM
Amadeo et al. [21]	MAC address in Data packet which is saved as an entry in FIB	–	Medium	Pedestrian /Vehicular	IEEE 802.11	ndnSIM
Abane et al. [22]	Cost field (eligibility of the node) in Interest and Data packets	Reinforcement learning-based timer before Interest broadcasting	Medium	IoT	IEEE 802.15.4	OMNeT++ /Arduino
Kuai and Hong [23]	Location information in Interest and Data packets	Distance-based collision-avoidance timer	Medium	VANETs	IEEE 802.11	ndnSIM
Our proposal	Node's ID in Interest packet, Node's ID and Hop count in Data packet	Distance&Energy aware Deferred-forwarding timer	High	IoT	IEEE 802.15.4	ndnSIM

limited memory of IoT nodes and the small MTU size of the communication protocols (Layer 2) to avoid packet fragmentation.

To this end, in our design, the FIB table contains, in addition to the prefix name and the output face, the ID of the next hop and its hop count (cost value) to the producer. The Interest packet carries, in addition to the original NDN fields, the ID of the next hop. Furthermore, as a final lightweight modification, the Data packet carries, in addition to the original NDN fields, the hop count to the producer and the ID of the sender. It is worth mentioning that, although we made a minor changes comparing to NDN basic design, in the aim to meet the constrained resources of the IoT nodes as explained above, the strength of our strategy is to manage this lightweight extra information, in the forwarding machinery, to provide an efficient and scalable learning mechanism, which will be explained in the subsequent subsections, permitting a smart collaboration between the IoT nodes to insure a better packet delivery with a minimum resource utilization.

Besides, for the proper description of the proposed strategy, the following assumptions are made:

- Nodes are constrained in terms of memory, computing, power, and communication capabilities in convenience with the IoT context;
- Nodes are equipped with a limited power source and are aware of their remaining energy;
- Each node has a unique 16 bits ID, which can be derived from its EUI 64 bits MAC address. Besides, all the nodes register to the common AllNodes_ID in order to receive broadcast packets.
- The mobility of the nodes is not continuous, they can be static for a certain period of time, which depends on the IoT application;
- Nodes communicate using the IEEE 802.15.4 standard and links are lossy;
- The adopted deployment mode in the proposed architecture is NDN overlay/IP [7]; and
- The considered communication mode in our proposed solution is the Pull-based one [7].

It is worthwhile to remember that in a NDN-based IoT ad hoc scenario, nodes have basically two faces, an *External Face* to exchange packets with other nodes and an *Internal Face* to exchange information with the up-layers, such as consumer and producer applications.

An illustrative example is given in Fig. 4, which shows the general functioning of LAFS before and after the learning mechanism. In this example, the inserted second line in the FIB table, for example for nodes 4 and 7, is the result of the updates issued from the provided learning mechanism of LAFS, which allows for the use of one forwarding path (C-7-8-9-P), for both Interest and

Data packets dissemination, instead of flooding the entire network.

The following describes the detailed LAFS approach and the mathematical formalization.

4.2 Detailed description of LAFS

In the sequel, we describe the whole machinery and characteristics of LAFS.

4.2.1 LAFS phases

LAFS operates according to the following phases.

1. *Deferred Interest Forwarding*: This first phase starts when an Interest packet visits for the first time a forwarder node. For this, a deferred timer is computed, after which, if the node does not receive the same Interest packet from one of its neighbors, it forwards the packet further, hop by hop, until the targeted Data provider is reached. Otherwise, the Interest forwarding is canceled and the packet is dropped. Our deferred Interest forwarding function is mainly based on the remaining energy of the forwarder and its hop count to the Data provider (this latter parameter is ignored if the Data provider is unknown). Therefore, LAFS allows reducing the broadcast storm phenomena, especially in the case of dense networks, as well as performing load balancing, energy efficiency, and timeliness delivery by making more eligible, forwarders having higher energy and closer to the Data providers.
2. *Data Forwarding and Learning*: The second phase of LAFS begins once a Data provider is reached by an Interest request. At this moment, the matched Data packet is forwarded, hop by hop, to the requester by following the reverse path taken by the Interest packet, thanks to the native stateful forwarding scheme of NDN.

The key advantage of LAFS is the learning process injected in the Data forwarding phase. Indeed, intermediate nodes involved in the Data dissemination path, including those located in the vicinity of the path (which have not a pending Interest corresponding to the received Data), store locally additional information about the previous node (from which the Data packet came), namely its ID and its hop count to the Data producer. These information are stored in the FIB table to set the corresponding Interest prefix as *Marked* and is updated automatically upon receiving a Data packet, carrying the same prefix, with a closer distance (hop count) to the producer. If the Interest prefix does not exist in the FIB table, it is implicitly considered as *Unmarked* Interest.

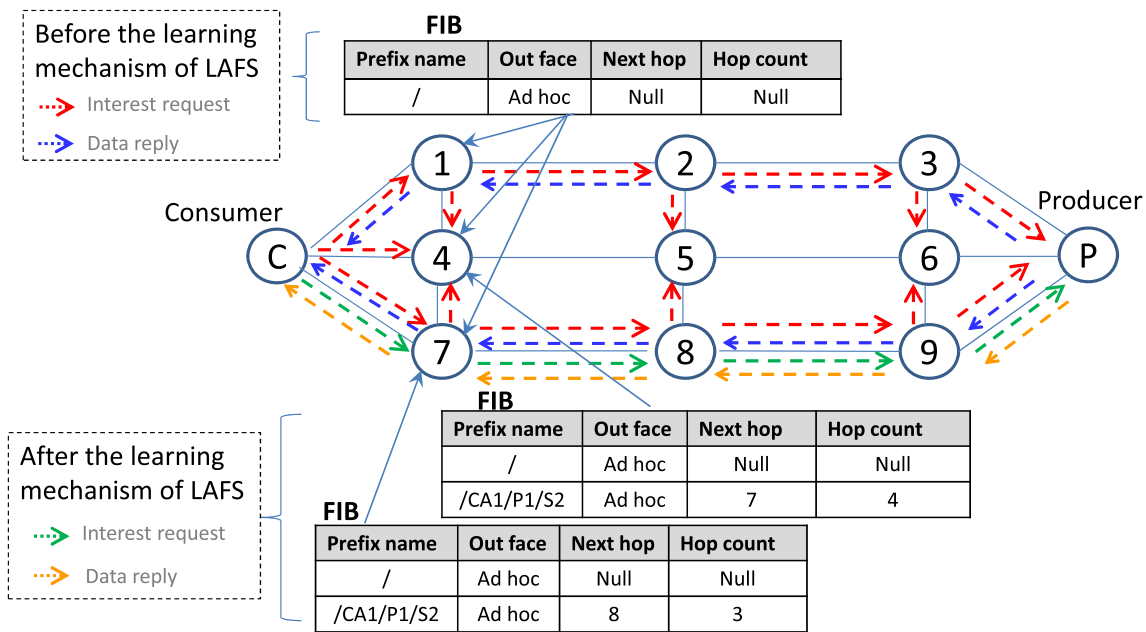


Fig. 4 Illustrative example of LAFS

These added lightweight operations allow building a *dynamic optimized forwarding path*, from the consumer to the producer, and make our LAFS strategy *adaptive* and *resilient* to potential changes that may occur to the network topology.

3. *Learning-based Interest Forwarding*: The third phase of LAFS permits the capitalization of the learning process performed in the second phase. When a marked Interest packet arrives at a node, whose prefix name exists in the FIB table, it is forwarded directly to the next hop, stored in the FIB, without triggering a waiting timer and so on hop by hop until reaching the Data provider. Upon arrival, the matching Data packet is natively forwarded to the consumer following the reverse path.

The objective of the *Learning-based Interest Forwarding* phase of LAFS is to retrieve as quickly as possible the searched Data for a marked Interest prefix, without wasting time and energy, by following a forwarding path that has been learned from the *Data Forwarding and Learning* phase.

4.2.2 Mobility support and smart energy management scheme provided by LAFS

As explained above, LAFS offers a smart and adaptive energy management scheme in a NDN-based IoT network with support for nodes mobility. Indeed, by including the remaining energy parameter in the deferred timer (refer to phase 1 of the strategy), this permits to dynamically favor nodes having a great amount of energy as eligible

forwarders. Moreover, LAFS allows building a path to the Data provider (refer to phase 3) through which the subsequent Interests having the same prefix name will be forwarded. So, to avoid energy depletion of the nodes belonging to this forwarding path, and overcome the problem of the mobility of the nodes, the built path will be valid for a certain time, during which the nodes are supposed to be static. After this period, the path is considered no longer valid, and LAFS returns to the *Deferred Interest Forwarding* phase, even for the marked Interest prefix names.

By doing so, various paths could be set up between the consumer and the producer, one at every path construction process, thus ensuring load balancing among the nodes of the IoT network. Also, the mobility of the intermediate nodes can be handled seamlessly, since the path is periodically rebuilt, after the end of validity time, and involves active and available nodes at every construction phase. Moreover, our strategy could be easily extended to register two or more next-hops and enable a more effective path construction technique with a robust fault-tolerance capability. Figure 5 presents the state-transition diagram of LAFS.

4.2.3 Deferred Interest forwarding function of LAFS

As we have seen earlier, in the *Deferred Interest Forwarding* phase, the eligibility of a node x to forward an Interest packet I with prefix p is based on a waiting time, which basically takes into consideration two factors: the energy of the node and its distance to the producer. The following

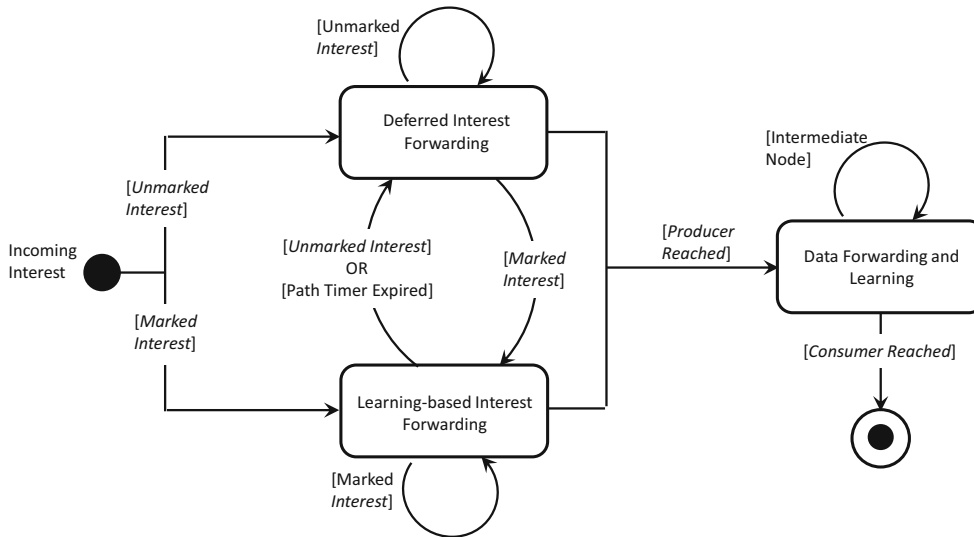


Fig. 5 State transition diagram of LAFS

equation is used to compute the waiting time a node observes before forwarding an Interest packet.

$$\phi_x(I(p)) = \begin{cases} \left[1 - \left(\alpha \frac{E_x}{E_i} + (1-\alpha) \frac{1}{HC_x(p)} \right) \right] T_{max}, & \text{if } I(p) \text{ is marked} \\ \left[1 - \left(\alpha \frac{E_x}{E_i} \right) \right] T_{max}, & \text{otherwise} \end{cases} \tag{1}$$

where E_x is the remaining energy of the node x and E_i its initial energy, $HC_x(p)$ is the Hop Count to the producer of the Data corresponding to the Interest $I(p)$, α is a weighting parameter between the energy and the hop count factors, and T_{max} is the maximum waiting delay a node can observe before sending an Interest packet.

This function ensures that the node having the best compromise between the remaining energy and the distance to the producer is the most eligible to forward the Interest packet among its neighbors. These neighbors cancel their Interest forwarding once they receive the same packet sent from the eligible node.

The algorithms used are detailed in the following subsections. All the notations and variables used in these algorithms are presented in Table 2.

4.2.4 Interest processing algorithm of LAFS

The Interest forwarding operation of our proposed solution works as follows.

Firstly, when an Interest packet comes to a node, it is inserted in the PIT table. After this, the verification process begins, resulting in canceling the forwarding operation or not, depending on the validity of the deferred forwarding timer. Afterward, the content store is checked to respond to the Interest request in case the correspondent Data is available. If not, the Interest prefix is checked in the

FIB, based on which either the deferred timer is triggered if the Interest prefix is marked, following the *Deferred Interest Forwarding Function* (1), or the Interest packet is directly forwarded to the next hop if it is previously marked.

It should also be noted that the node uses direct forwarding in case of incoming Interest from the internal face (application face of the consumer), even if that Interest packet is not marked before. Algorithm 1 summarizes the Interest forwarding process of LAFS, which encompasses both phase 1 and phase 3.

Table 2 Notations and variables used in algorithms of LAFS

Parameter	Description
Interest(p)	Interest packet with prefix: p
Face(f)	Face having identifier (ID): f
PIT(p).Timer	Deferred Interest Forwarding Timer for PIT entry of prefix(p)
Data(p)	Data packet correspondent to prefix(p)
Interest(p).NextHop	The field: <i>next hop</i> of Interest(p)
FIB(p)	FIB entry for prefix(p)
Pending_Interest(p)	Pending Interest(p) in PIT
FIB(p).NextHop	The field: <i>next hop</i> of FIB(p)
Information(Data(p))	Carried information of Data(p): <i>ID</i> of the sender, <i>hop count</i> to the Data producer
FIB(p).HopCount	The field: <i>hop count</i> of FIB(p)
Data(p).HopCount	The field: <i>hop count</i> of Data(p)
FIB(p).PathTimer	The Path validity timer for FIB(p)
AllNodes_ID	16-bit constant that represents all nodes' IDs (network layer)
Node_ID	The ID of the current Node

Algorithm 1 Interest processing algorithm LAFS.

```

1: procedure ONINCOMINGINTEREST(Prefix:  $p$ , InFace:  $f$ )
2:   Insert [ $Interest(p)$ ,  $Face(f)$ ] in  $PIT$ ;
3:   if ( $Interest(p)$  is Duplicated) then                                     ▷ Receive duplicate  $Interest(p)$ 
4:     if ( $PIT(p).Timer$  is Active) then                                     ▷ Within deferred forwarding delay
5:       Cancel  $PIT(p).Timer$ ;                                           ▷ Cancel sending the Interest
6:     else
7:       Delete Duplicated [ $Interest(p)$ ,  $Face(f)$ ];                       ▷ Ignore the duplicated Interest
8:     end if
9:   Return;
10: end if
11: if ( $CS$  contains  $Data(p)$ ) then                                         ▷ Cache hit
12:   Send  $Data(p)$  to  $Face(f)$ ;                                             ▷ Respond to the  $Interest$  with matching  $Data$ 
13:   Delete [ $Interest(p)$ ,  $Face(f)$ ] from  $PIT$ ;
14:   Return;
15: end if
16: if ( $Interest(p)$  is Unmarked) AND ( $Face(f)$  is External) then         ▷ Interest coming from External Node
17:   if ( $Interest(p).NextHop == AllNodes\_ID$ ) OR
18: ( $Interest(p).NextHop == Node\_ID$ ) then
19:     Execute ( ) for  $PIT(p).Timer$ ;                                       ▷ Deferred Interest Forwarding
20:     if ( $PIT(p).Timer$  is Fired) then                                     ▷ On expiry Timer
21:       if ( $Pending\_Interest(p)$ ) then                                     ▷ Check condition on  $PIT(p).Timer$  Expiry
22:          $Interest(p).NextHop := AllNodes\_ID$ ;
23:         Send  $Interest(p)$ ;                                             ▷ Broadcast the Interest
24:       end if
25:     end if
26:   end if
27: else
28:   if [( $Interest(p)$  is Marked) AND ( $Face(f)$  is Internal)] OR
29: [( $Face(f)$  is External) AND ( $Interest(p).NextHop == Node\_ID$ )] then
30:    $Interest(p).NextHop := FIB(p).NextHop$ ;
31:   Send  $Interest(p)$ ;
32: else
33:   if ( $Face(f)$  is Internal) then                                         ▷ Case of App_Face of the Consumer
34:      $Interest(p).NextHop := AllNodes\_ID$ ;
35:     Send  $Interest(p)$ ;
36:   end if
37: end if
38: end if
39: end procedure

```

4.2.5 Data processing algorithm of LAFS

The Data forwarding processing of LAFS works as follows.

On incoming Data packet, the node verifies first if there is a correspondent pending Interest in its PIT. If there is not, it updates its FIB entry by the carried information of the Data packet (ID and hop count) before dropping that packet; we call this opportunistic dynamic updates of the learning phase. In the case of a valid PIT entry for the incoming Data, this latter is stored in the CS before processing it.

In doing so, if the Data packet comes from an external face (neighbor node), the receiver marks the correspondent prefix name in the FIB, saves the next hop and the hop count to the Data producer, and triggers the path validity timer for this prefix name. Subsequently, the Data packet is forwarded to all incoming faces in the correspondent PIT entry after updating the embedded information, namely the hop count which is incremented, and the next hop which is replaced by the actual forwarder ID. The PIT entry is systematically deleted once the Data packet is sent.

Algorithm 2 Data processing algorithm of the proposed solution

```

1: procedure ONINCOMINGDATA(Prefix:  $p$ , InFace:  $f$ )
2:   if (Not Pending_Interest( $p$ )) then                                ▷ The nodes on the path vicinity
3:     if ( $FIB(p).HopCount == NULL$ ) then
4:       Insert Information(Data(p)) in  $FIB(p)$ ;                                ▷ Add carried information of
       Data(p) ( $p, f, HopCount, NextHop : NULL$ ) to  $FIB(p)$ 
5:     else
6:       if ( $FIB(p).HopCount > Data(p).HopCount$ ) then
7:         Update  $FIB(p)$  with Information(Data(p));                                ▷ Update if better distance
8:       end if
9:     end if
10:    Return;                                                                ▷ Ignore incoming Data packet
11:  end if
12:  Store Data(p) in  $CS$ ;                                                    ▷ Cache placement of Data Payload
13:  if (Face( $f$ ) is External) then                                        ▷ In case Data comes from neighbor node
14:    if ( $FIB(p).NextHop == NULL$ ) then                                    ▷ Interest(p) not marked yet
15:      Insert Information(Data(p)) in  $FIB$ ;                                ▷ Insert ( $p, f, HopCount, NextHop$ ) to
       $FIB$  to mark Interest(p)
16:       $FIB(p).Path\_Timer.Start()$ ;                                ▷ Trigger Path validity Timer for Interest(p)
17:      if ( $FIB(p).Path\_Timer$ ) is Fired then                                ▷ On expiry Path_Timer
18:         $FIB(p).NextHop := NULL$  ;                                ▷ Unmark Interest(p) to reset the path
19:      end if
20:    end if
21:     $Data(p).HopCount := Data(p).HopCount + 1$ ;
22:  else                                                                    ▷ In case Data comes from the application layer (Internal Face)
23:     $Data(p).HopCount := 0$ ;                                            ▷ Start hop count from the Producer
24:  end if
25:   $Data(p).NextHop := Node\_ID$ ;
26:  Send Data(p) to Pending_Interest(p)Faces in  $PIT$ ;
27:  Delete Interest(p)Entry from  $PIT$ ;                                    ▷ Along with all Incoming Faces
28: end procedure

```

Moreover, on the expiration of the path validity timer, the correspondent next hop in the FIB is reset to the default value, i.e., “NULL”, and the Interest prefix name is considered unmarked. Algorithm 2 details the Data forwarding process of LAFS (phase 2).

5 Performance evaluation

5.1 Use case: a smart healthcare monitoring system

In our case study, we are interested in automatic sensor-based medical surveillance of suspicious COVID-19 infections, during the period of preventive confinement. The goal is to provide patients with permanent and real-time remote monitoring of their health to allow fast and accurate interventions while minimizing physical contact.

5.1.1 Design and functioning of the system

The design of our system is described below. Suspicious patients are placed in a building (or a hotel) in individual rooms. Each patient wears two types of battery-powered autonomous sensors:

- *Sensor1*: To measure body temperature and monitor fever (worn on the wrist);
- *Sensor2*: To measure the patient’s respiratory rate, by detecting the variation in the circumference of the abdomen [32] (a belt placed at the waist).

Both sensors use a single IEEE 802.15.4 interface for low-power wireless communication. Hence, a patient is seen as an IoT node dotted with a communication interface, and capable of measuring two types of data (temperature and respiratory rate).

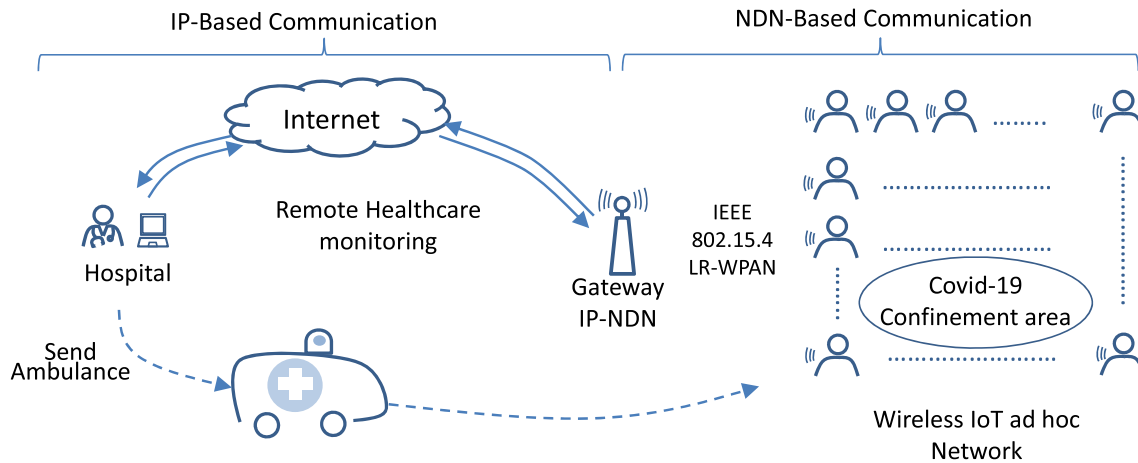


Fig. 6 The proposed Smart Healthcare Monitoring System

From a remote hospital, a medical team continuously monitors the health of patients, by collecting over the Internet and analyzing data measured by worn sensors. To this end, a gateway is placed near the confinement zone and equipped with a dual network; an IEEE 802.15.4 communication interface and a standard IP interface. Once the symptoms related to COVID-19 are observed, an ambulance is sent to evacuate the patient to the hospital (see Fig. 6).

5.1.2 Main system characteristics

Our system has the following characteristics in terms of architecture, infrastructure, and communication mode.

- **Architecture:** Since the main purpose of this system is the collection of the data produced by the patients' sensors, we have opted for the use of a data-oriented architecture following the NDN underlay IP model. The native NDN network is deployed locally on IoT devices (in the confinement zone), and is connected to the global Internet through an IP/NDN gateway.
- **Infrastructure:** In the confinement zone, IoT nodes are deployed in an ad hoc wireless infrastructure. These nodes are identified by patients' identifiers and their locations are not known beforehand.
- **Mode of communication:** Communication in the confinement zone is based on hop-by-hop packet transmission, where two modes are supported:
 - **PULL mode:** The doctor, considered as a consumer, periodically monitors the state of health of patients (producers), by sending them a generic Interest message (intended for all patients) or a targeted one (intended for a given patient). The patients concerned by the request respond with a Data message containing the

measurement of the medical sensor specified in the Interest (Sensor1 or Sensor2);

- **PUSH mode:** This mode is triggered if the configured thresholds of the sensor measurements are reached. In such a case, without waiting for an Interest, the patient's node systematically sends a Data message to the doctor to alert him of his condition.
- **Naming scheme:** Our naming system adopts a lightweight hierarchical and human-readable naming structure supportable by resource-constrained IoT nodes. It is composed of three fields delimited by "/" as follows *"/ConfinementArea/Patient/Sensor"*, where:
 - The root *"ConfinementArea"* identifies the targeted confinement zone, for example *"CA1"*;
 - The field *"Patient"* represents the identifier of the patient in confinement, for example *"P1"*;
 - The *"Sensor"* field specifies the type of the medical sensor, *"S1"* or *"S2"*.

Thus, the consumer (doctor) sends an Interest bearing the hierarchical name as defined above, for example, *"/CA1/P1/S2"*. The producer (patient) responds with the measurement of the sensor specified in the request, respecting the NDN naming system, i.e., the Data message consists of the hierarchical name sought, the corresponding Data, and its crypto-signature. Finally, it should be noted that our proposal is extensible and allows the addition of other types of sensors and/or actuators.

5.2 Evaluation platform and settings

For the sake of this evaluation, we have implemented LAFS in ndnSIM 2.8 using the IEEE 802.15.4 communication

Table 3 Simulation parameters

Parameter	Value	Parameter	Value
NetDevice	LrWPAN (IEEE 802.15.4)	Number of consumers	1–5
Area size ($m \times m$)	50×50–90×90	Number of producers	1–5
Number of nodes	36–100	CS size	10
Topology	Grid	PIT size	10
Initial Energy (j)	5	Interest packet size (<i>bytes</i>)	5
Simulation time (s)	100–600	Data packet size (<i>bytes</i>)	10
Interest Transmission Rate (<i>packet/s</i>)	1, 5, 10, 15, 20	Deferred forwarding parameters (1)	Factor α T_{max} (<i>ms</i>) 20
Transmission Range (m)	10, 30, 50, 70, 90	Path expiry time (s)	10

standard as an underlay to the NDN layer. We have used two grid topologies of 36 and 100 nodes while varying the Interest transmission rate, the transmission range, the number of consumers and producers as shown in Table 3.

5.3 Performance metrics

Our evaluations aim to highlight the strengths and weaknesses of the compared schemes under a wide range of simulation settings. Specifically, the following performance metrics are considered:

- *Sent Interests*: The total number of Interest packets that have been forwarded in the network;
- *Sent Data*: The total number of Data packets that have been forwarded in the network;
- *Success rate*: The average success rate of all consumers (satisfied Interests), which is measured as the total number of received Data packets divided by the total number of sent Interest packets;
- *Hop count*: The average hop count for all Data packets;
- *Retrieval time*: The average retrieval time of all Data packets; and
- *Remaining energy*: The average remaining energy of all the nodes, which is expressed as a percentage of the initial network energy. For this, we use the energy model of ns-3 [33].

Simulation results are averaged over 10 independent runs. The mean along with the correspondent 95% confidence intervals are reported in the graphs. Besides, the performance evaluation was conducted on two batteries of extensive simulations. The former assesses the impact of each mechanism separately on LAFS performance to pinpoint the potential enhancements of each mechanism on the network, whereas the latter battery is devoted to perform a comprehensive evaluation of the whole LAFS solution against competitive protocols.

5.4 Evaluation of LAFS mechanisms

Through this evaluation, we intend to assess the impact of each of the main features of LAFS on the network performance, namely the Deferred Interest Forwarding scheme (DIF), which constitutes the first LAFS phase and the Learning Mechanism (LM), which combines phases two and three of LAFS (Section 4.2.1). To this end, we compare the final version of LAFS against LAFS-DIF and LAFS-LM. A grid topology of 36 nodes has been used in this configuration with one consumer and one producer. The transmission range was fixed at 10 m, the simulation time was set to 100 s, and the Interest transmission rate was varied from 1 to 20 p/s. Figure 7 reports obtained results in terms of success rate and retrieval time.

It is clear from the results depicted in Fig. 7a that the gains in retrieval times are mainly influenced by the LAFS learning mechanism. Indeed, the learning process of LAFS-LM permitted to build the paths to be borrowed by Interest packets, which avoided the per-hop waiting times of LAFS-DIF and allowed reaching the content providers as quickly as possible. Precisely, content retrieval time values were nearly halved by LAFS and LAFS-LM when compared to LAFS-DIF under light load conditions (1 p/s). This gain increased with increasing traffic load augments (around 75% reduction of retrieval time for 20 p/s). This witnesses the higher scalability of LAFS-LM.

On the other hand, the average success rate is mainly affected by the traffic overload in the network, as depicted in Fig. 7b. Overall, all evaluated solutions showed similar success rates under light traffic loads, with LAFS-DIF achieving slightly better rates at 5 p/s thanks to the reduction in redundant packet transmissions by allowing only eligible nodes to forward the Interest. For higher traffic loads, a case-dependent success rate trade-off between LAFS-DIF and LAFS-LM was registered. Thus, while LAFS-DIF showed better rates at 20 p/s thanks to minimized redundant

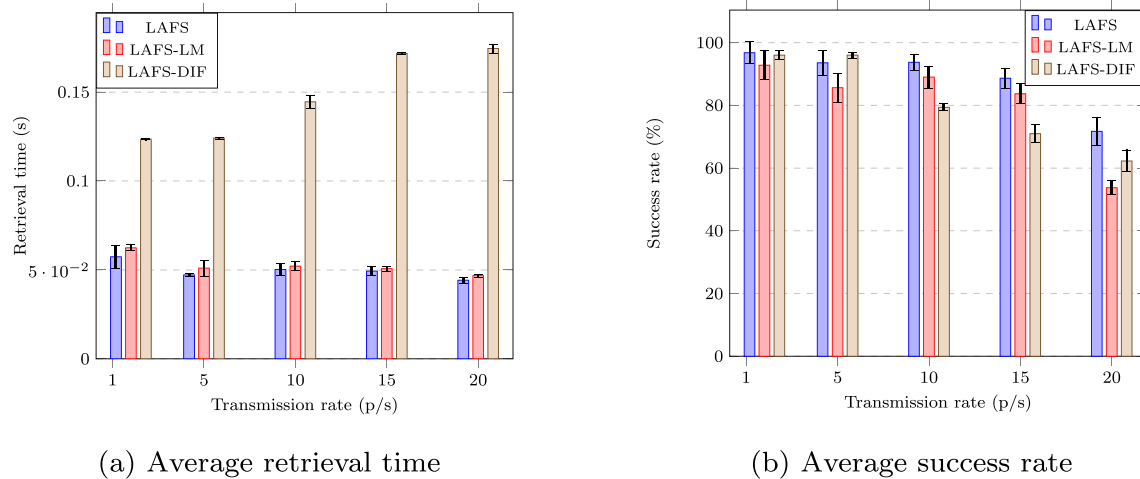


Fig. 7 Impact of the inherent features of LAFS. **a** Average retrieval time. **b** Average success rate

transmissions, LAFS-LM registered better success rates at 10 and 15 p/s content request frequencies thanks to the use of unique learned paths for marked Interests. In all cases, LAFS achieved the best success rates due to the combination of both mechanisms.

In summary, this first round of simulations reveals that the combination of both LAFS mechanisms permits to reduce content retrieval time while ensuring a high request satisfaction ratio.

5.5 Evaluation of LAFS against other solutions

To evaluate the performance of the proposed solution LAFS, we compare it with the following forwarding protocols: (i) a Blind Deferred Interest Forwarding (BDF), inspired from [15], which employs collision avoidance timers, based on random delays, for Interest packets dissemination, with no awareness about the network topology; and (ii) the Native Forwarding mechanism of NDN without extensions, which we call (NF). To do so, multiple simulations were run and different results were collected to understand how the Interest demand frequency, the network density, and the number of consumers and producers affect the behavior of the evaluated solutions. To this end, a larger topology has been used, which is composed of 100 nodes deployed in a 90×90 m² area, and all the metrics were measured after running 600 seconds of simulation time.

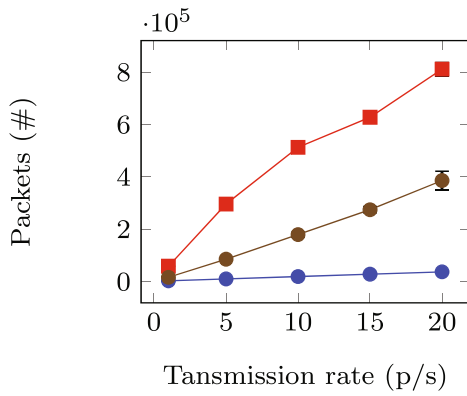
5.5.1 Impact of the Interest transmission rate

For this first simulation set, we have fixed the transmission range of all nodes to 50 m (including one producer and one consumer), varied the Interest frequency from 1 to 20 p/s, and measured the performance metrics of Section 5.3.

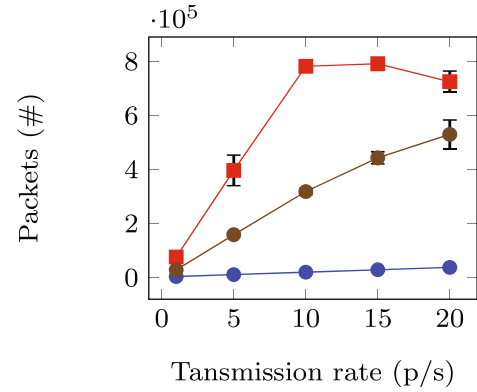
Results depicted in Fig. 8 show that LAFS outperformed BDF and NF for almost all the observed network metrics. Indeed, Fig. 8a and b, which reflect the total number of exchanged Interest and Data packets, respectively, show that LAFS generated around 95% and 91% fewer packets compared to NF and BDF respectively, in the highest request frequency of 20 p/s. Concerning success rate, LAFS has preserved a high satisfaction ratio for around 98% despite the increase of the Interest demands frequency, whereas it has been considerably degraded in case of NF and BDF, registering a value between 50 and 60% for a demand frequency of 20 p/s. This performance is mainly due to the LAFS learning process.

Regarding the average content retrieval time and remaining energy, Figs. 8d and f indicate, respectively, that LAFS offers a faster and more energy-efficient way to retrieve Data from the producer than NF and BDF. Indeed, the content retrieval time has been shortened by LAFS for around 93% and 97% compared to BDF and NF, respectively, while LAFS saved about 81% and 78% energy more than BDF and NF. This higher performance is again the result of LAFS mechanisms, which avoided the broadcast storms.

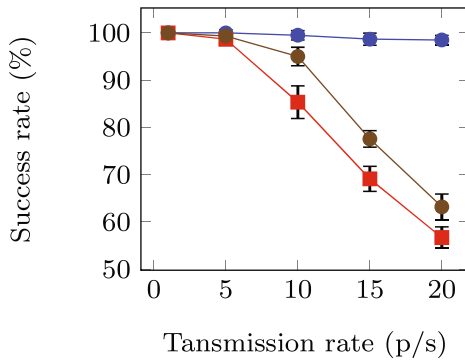
Finally, the average hop count of LAFS (Fig. 8e) is better than the other two, despite the use of a single forwarding path issued from the learning mechanism. Indeed, for both NF and BDF protocols the path length is proportional to the Interest demand frequencies because of the induced traffic overload. Nevertheless, LAFS does not fall in the same situation where the path length is almost stabilized independently of the Interest demand frequency, which allows to reduce up to half the overall hop count for a transmission rate of 20 p/s. This is due to the adopted forwarding eligibility function of LAFS that



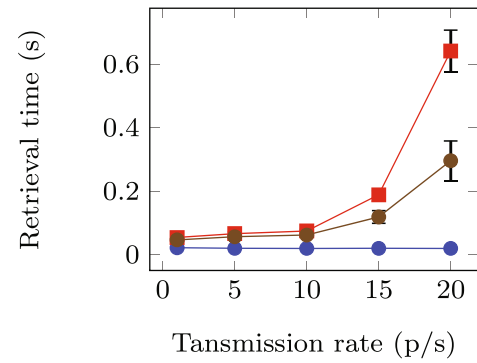
(a) Sent Interest packets



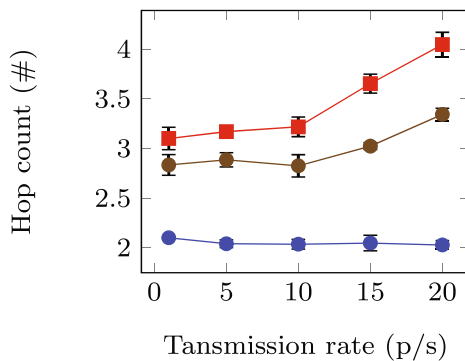
(b) Sent Data packets



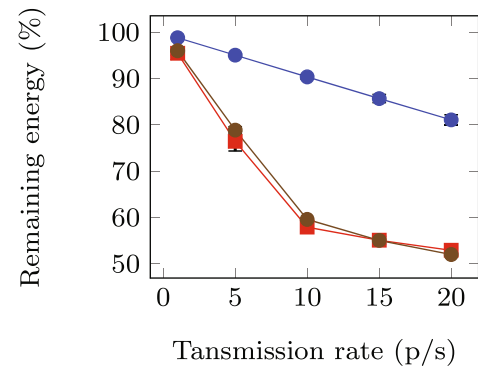
(c) Average success rate



(d) Average retrieval time



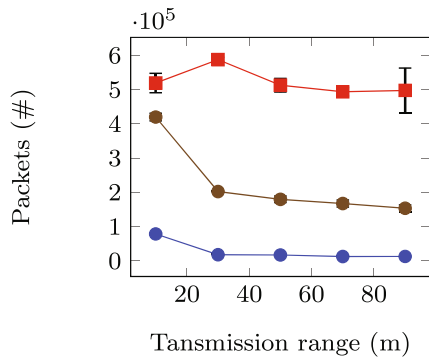
(e) Average hop count



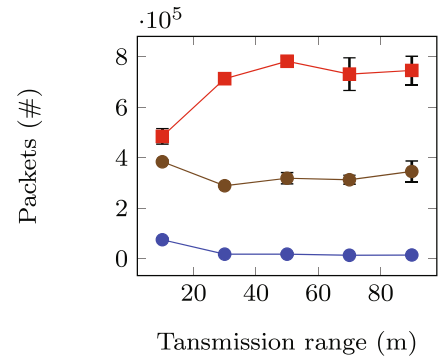
(f) Average remaining energy



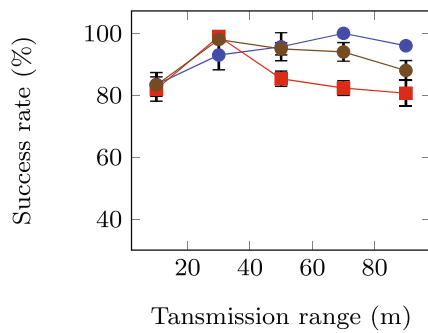
Fig. 8 Impact of the Interest transmission rate. **a** Sent Interest packets. **b** Sent Data packets. **c** Average success rate. **d** Average retrieval time. **e** Average hop count. **f** Average remaining energy



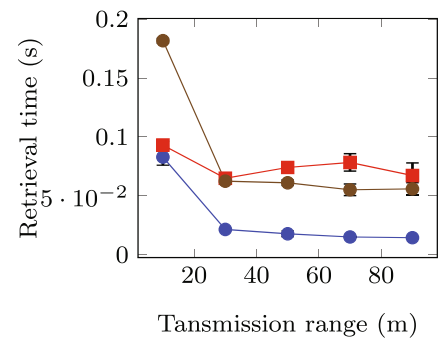
(a) Sent Interest packets



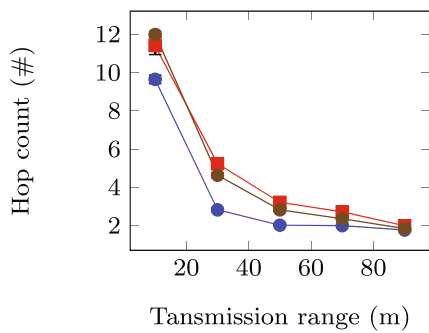
(b) Sent Data packets



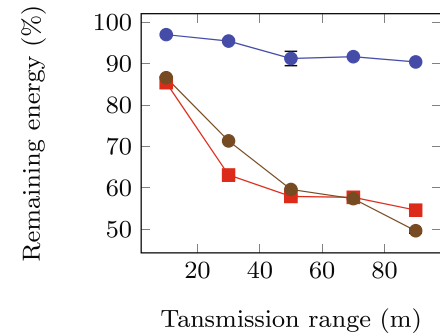
(c) Average success rate



(d) Average retrieval time



(e) Average hop count



(f) Average remaining energy



Fig. 9 Impact of the transmission range. **a** Sent Interest packets. **b** Sent Data packets. **c** Average success rate. **d** Average retrieval time. **e** Average hop count. **f** Average remaining energy

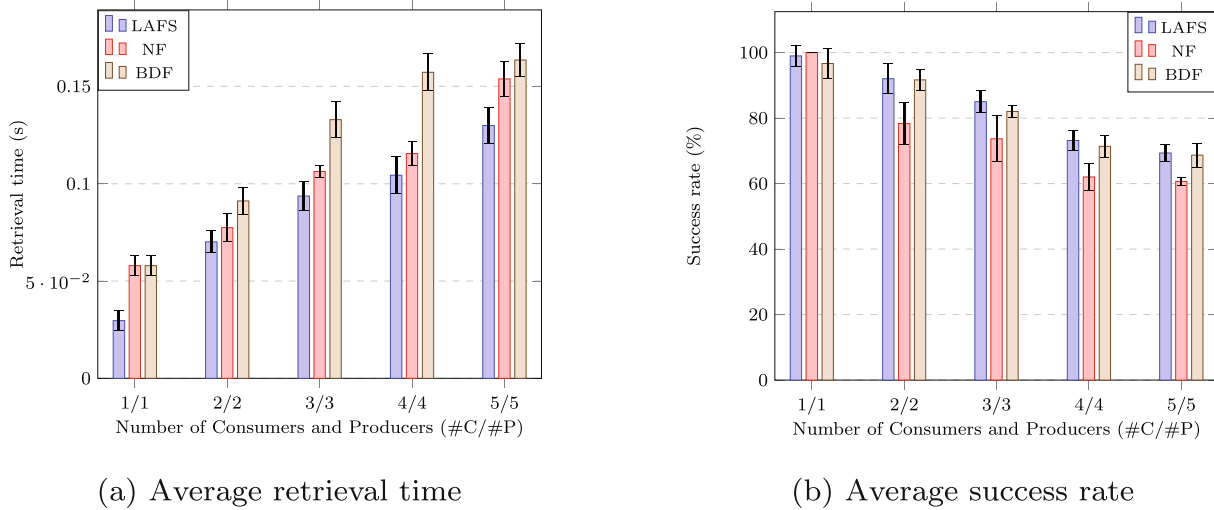


Fig. 10 Impact of the number of consumers and producers. **a** Average retrieval time. **b** Average success rate

takes into account, in addition to the remaining energy of the node, its hop count to the Data provider, which favor the closest nodes to the producer to be more eligible forwarders.

5.5.2 Impact of the nodes’ transmission range

Throughout this second set of simulations, we intend to assess the impact of network density on the benchmarked solutions. To this end, we have fixed the Interest transmission rate to 10 p/s, and varied the transmission range of the nodes from 10 to 90 m. One producer and one consumer were set up in this configuration. The obtained results are shown in Fig. 9.

These results clearly show that under almost all tested network densities LAFS significantly outperforms the

two other forwarding solutions, which were penalized by the repeated broadcast operations, whether deferred or not, especially in the case of dense networks, where a great number of nodes fall in the same broadcast domain. Consequently, BDF and NF incurred extra energy consumption and extended response times compared to LAFS, as shown in Fig. 9f and d, respectively. On the other hand, the controlled broadcast operations of LAFS allowed a reduction in Interest (re)transmissions and collisions in a dense network, as depicted in Fig. 9a and b. Also, the hop count factor used in our forwarding function led to building an optimized path to the Data producer, hence, ensuring an average hop count to the Data providers slightly better than the two other strategies (see Fig. 9e).

On the other side, as a little drawback of LAFS, we noticed a performance hit compared to NF and BDF,

Table 4 Main findings about the behavior of the benchmarked solutions

Parameter	Value	Success rate			Retrieval time			Forwarded packets			Hop count			Energy efficiency		
		NF	BDF	LAFS	NF	BDF	LAFS	NF	BDF	LAFS	NF	BDF	LAFS	NF	BDF	LAFS
Interest demand frequency	Low	++	++	++	++	++	++	+	++	++	+	+	++	+	++	++
	High	-	-	++	-	+	++	--	-	++	-	+	++	--	--	+
Transmission range	Low	++	++	+	+	-	+	-	-	++	-	-	+	+	+	++
	High	+	+	++	+	+	++	--	-	++	+	+	++	-	-	++
Number of consumers/producers	Low	++	++	++	+	+	++									
	High	-	+	+	-	-	+									

++: Good
 +: Fair
 -: Poor
 --: Very poor

concerning the request success rate metric in case of values of transmission range inferior to 40 m (see Fig 9c). This can be explained by the unreliable nature of the wireless communication medium, which incurs packet loss. Indeed, as the transmission range of the forwarder node decreases, the probability of its neighbor to receive a forwarded packet decreases too. Consequently, one path forwarding scheme of LAFS will be penalized in terms of packet success rate compared to BDF and NF, both of which use multipath forwarding. Nevertheless, LAFS registered a better success rate for higher transmission range values (above 40 m), and this outperformance becomes more tangible as the transmission range increases, thanks to the provided learning mechanism.

5.5.3 Impact of the number of producers/consumers

This third set of experiment was conducted to evaluate the impact of the number of producers and consumers on the performance of the compared solutions. To this end, the transmission range of the nodes was fixed at 30 m, the Interest transmission frequency was set to 1 p/s, and the number of consumers/producers was simultaneously varied from 1 to 5. Two metrics were measured in this configuration, namely the success rate and the retrieval time provided by each strategy, which are reported in Fig. 10.

Obtained results confirm that LAFS outperforms NF and BDF for the both measured metrics due to the same reasons mentioned above, namely the learning mechanism and the adaptive deferred Interest forwarding function. Indeed, these two mechanisms permitted to handle efficiently the network traffic caused by multiple Interest requests and Data replies.

Once it comes to individual metrics, the average content retrieval time increased with increasing number of consumers/producers for the three solutions, as depicted in Fig. 10a. This is due to the fact that increasing the number of producers and consumers raised the overhead in the network, which led to extending packet waiting periods at the forwarders. With regards to the success rate metric, Fig. 10b shows that increasing the number of consumers and producers resulted in decreasing request satisfaction rates. This is due to the huge number of forwarded Interests and Data packets within a short period, which resulted in more packet collisions and negatively affected the retrieval content process of the compared strategies.

To sum up, the carried-out simulations show that our proposed forwarding scheme LAFS, offers an efficient solution to counteract the well-known broadcast problem in NDN-based wireless ad hoc IoT networks. Moreover, the impact of the Interest demands frequency, the network density, and the number of producers and consumers on the behavior of the compared solutions reveal that

LAFS better deals with network resources saving than the other protocols (NF and BDF) while guarantying efficiency and effectiveness, especially in terms of success rate, retrieval time, and remaining energy in the IoT environment.

Table 4 summarizes the main findings about the behavior of NF, BDF, and LAFS under different settings.

6 Conclusion and future work

This paper presented a new forwarding strategy tailored for NDN-based wireless ad hoc IoT networks. After highlighting the particularities of NDN in dealing with wireless ad hoc environments, existing solutions have been discussed. This has been followed by proposing a new Learning-based Adaptive Forwarding Strategy (LAFS) for NDN-based IoT networks, which incorporates a learning process along with an adaptive forwarding eligibility function. The provided mechanisms enable smart cooperation between nodes to handle dynamically main forwarding issues in this wireless constrained environment. We have evaluated and compared the proposed scheme with two representative solutions. The obtained results confirm that our strategy outperforms the other ones, allowing reducing the broadcast storm problem as well as performing energy-efficient and timely delivery in a constrained-resources IoT deployment.

In the future, we intend to evaluate the proposed solution in a more complex IoT scenarios, including nodes' mobility, fault tolerance, and push communication mode handling, along with an implementation on a real IoT platform.

References

1. Djama A, Djamaa B, Senouci MR (2019) Tcp/ip and ics networking technologies for the internet of things: a comparative study. In: The 4th international conference on networking and advanced systems (ICNAS). IEEE, Annaba, pp 1–6
2. Jacobson V, Smetters DK, Thornton JD, Plass MF, Briggs NH, Braynard RL (2009) Networking named content. In: Proceedings of the 5th international conference on Emerging networking experiments and technologies. ACM, pp 1–12
3. Boukerche A, Turgut B, Aydin N, Ahmad MZ, Bölöni L, Turgut D (2011) Routing protocols in ad hoc networks A survey. *Comput Netw* 55(13):3032–3080
4. Shu L, Zhang Y, Yang LT, Wang Yu, Hauswirth M, Xiong N (2010) Tpgf: geographic routing in wireless multimedia sensor networks. *Telecommun Syst* 44(1-2):79–95
5. Bouatit MN, Boumerdassi S, Minet P, Djama A (2016) Fault-tolerant mechanism for multimedia transmission in wireless sensor networks. In: 2016 IEEE 84th vehicular technology conference (VTC-Fall), pp 1–6
6. Tseng Y-C, Ni S-Y, Chen Y-S, Sheu J-P (2002) The broadcast storm problem in a mobile ad hoc network. *Wirel. Netw.* 8(2/3):153–167

7. Djama A, Djamaa B, Senouci MR (2020) Information-centric networking solutions for the internet of things: a systematic mapping review. *Comput Commun* 159:37–59
8. NDN Forwarder Daemon. <https://named-data.net/doc/NFD/current/> [Accessed Jan. 21, 2021]
9. Zhang L, Afanasyev A, Burke J, Jacobson V, Crowley P, Papadopoulos C, Wang L, Zhang B et al (2014) Named data networking. *ACM SIGCOMM Comp Comm. Rev* 44(3):66–73
10. Lipman J, Liu H, Stojmenovic I (2009) Broadcast in ad hoc networks. In: *Guide to wireless ad hoc networks*. Springer, pp 121–150
11. Wang L, Afanasyev A, Kuntz R, Vuyyuru R, Wakikawa R, Zhang L (2012) 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, NoM '12*. Association for Computing Machinery, New York, pp 7–12
12. Angius F, Gerla M, Pau G (2012) Bloogo: Bloom filter based gossip algorithm for wireless ndn. In: *Proceedings of the 1st ACM workshop on emerging name-oriented mobile networking design - architecture, algorithms, and applications, NoM '12*. Association for Computing Machinery, New York, pp 25–30
13. Amadeo M, Campolo C, Molinaro A, Mitton N (2013) Named data networking: A natural design for data collection in wireless sensor networks. In: *2013 IFIP wireless days (WD)*. pp 1–6
14. Intanagonwiwat C, Govindan R, Estrin D, Heidemann J, Silva F (2003) Directed diffusion for wireless sensor networking. *IEEE/ACM Trans Netw (ToN)* 11(1):2–16
15. Amadeo M, Molinaro A, Ruggeri G (2013) E-chanet: Routing, forwarding and transport in information-centric multihop wireless networks. *Comput Commun* 36(7):792–803
16. Yu Y, Dilmaghani RB, Calo S, Sanadidi MY, Gerla M (2013) Interest propagation in named data manets. In: *2013 international conference on computing, networking and communications (ICNC)*. pp 1118–1122
17. You Lu, Zhou B, Tung Lung-Chih, Gerla M, Ramesh A, Nagaraja L (2013) Energy-efficient content retrieval in mobile cloud. *Association for Computing Machinery, New York*, pp 21–26
18. Baccelli E, Mehlis C, Hahm O, Schmidt T, Wählisch M (2014) Information centric networking in the IoT: experiments with NDN in the wild. In: *1st ACM conference on information-centric networking (ICN-2014)*. ACM, Paris
19. Aboud A, Touati H, Hnich B (2019) Efficient forwarding strategy in a ndn-based internet of things. *Clust Comput* 22(3):805–818
20. Gao S, Zhang H, Zhang B (2016) Energy efficient interest forwarding in ndn-based wireless sensor networks. *Mob Inf Syst* 2016
21. Amadeo M, Campolo C, Molinaro A (2017) A novel hybrid forwarding strategy for content delivery in wireless information-centric networks. *Comput Commun* 109:104–116
22. Abane A, Daoui M, Bouzeffrane S, Muhlethaler P (2019) A lightweight forwarding strategy for named data networking in low-end iot. *J Netw Comput Appl* 148:102445
23. Kuai M, Hong X (2019) Location-based deferred broadcast for ad-hoc named data networking. *Future Internet* 11(6)
24. ndnSIM Simulator, 08 2020. <https://ndnsim.net/current/>
25. CCN-lite Project. <http://ccn-lite.net/> [Accessed Dec. 20, 2020]
26. RIOT operating system. <http://www.riot-os.org/> [Accessed Jan. 21, 2021]
27. QualNet network simulation software. <https://www.scalable-networks.com/products/qualnet-network-simulation-software-tool/> [Accessed Dec. 20, 2020]
28. Common Open Research Emulator. <https://www.nrl.navy.mil/itd/ncs/products/core> [Accessed Dec. 20, 2020]
29. Issariyakul T, Hossain E (2009) Introduction to network simulator 2 (ns2). In: *Introduction to network simulator NS2*. Springer, pp 1–18
30. OMNeT++. <https://omnetpp.org/> [Accessed Jan. 21, 2021]
31. Arduino. <https://www.arduino.cc/> [Accessed Jan. 21, 2021]
32. He Z, Zhang J, Hu Z, Quan L, Shi L, Chen J, Xuan W, Zhang Z, Dong S, Luo J (2019) Waist-wearable wireless respiration sensor based on triboelectric effect. *Nano Energy* 59:75–83
33. The Network Simulator. <https://www.nsnam.org/> [Accessed Dec. 20, 2020]

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.