

Article

# ndnIoT-FC: IoT Devices as First-Class Traffic in Name Data Networks

**Luís Gameiro** <sup>1</sup>, **Carlos Senna** <sup>1,\*</sup>  and **Miguel Luís** <sup>1,2</sup> <sup>1</sup> Instituto de Telecomunicações, 3810-193 Aveiro, Portugal; luismiguelgameiro@ua.pt (L.G.); nmal@av.it.pt (M.L.)<sup>2</sup> ISEL—Instituto Superior de Engenharia de Lisboa, 1959-007 Lisboa, Portugal

\* Correspondence: cr.senna@av.it.pt

Received: 15 October 2020; Accepted: 20 November 2020; Published: 21 November 2020



**Abstract:** In recent years we have been assisting a radical change in the way devices are connected to the Internet. In this new scope, the traditional TCP/IP host-centric network fails in large-scale mobile wireless distributed environments, such as IoT scenarios, due to node mobility, dynamic topologies and intermittent connectivity, and the Information-Centric Networking (ICN) paradigm has been considered the most promising candidate to overcome the drawbacks of host-centric architectures. Despite bringing efficient solutions for content distribution, the basic ICN operating principle, where content must always be associated with an interest, has serious restrictions in IoT environments in relation to scale, performance, and naming, among others. To address such drawbacks, we are presenting ndnIoT-FC, an NDN-based architecture that respects the ICN rules but offers special treatment for IoT traffic. It combines efficient hybrid naming with strategies to minimize the number of interests and uses caching strategies that virtually eliminates copies of IoT data from intermediate nodes. The ndnIoT-FC makes available new NDN-based application-to-application protocol to implement a signature model operation and tools to manage its life cycle, following a publisher-subscriber scheme. To demonstrate the versatility of the proposed architecture, we show the results of the efficient gathering of environmental information in a simulation environment considering different and distinct use cases.

**Keywords:** Internet of Things; Smart City; Information-Centric Network; Named Data Networks; publish-subscribe

---

## 1. Introduction

The Internet of Things (IoT), a recent communication paradigm in which objects of everyday life are able to, among others, communicate with one another, has become an integral part of the Internet [1,2]. The IoT devices have special characteristics, such as heterogeneous capacities (power, memory, battery) and in general these devices are small, constraint-oriented wireless sensors, and of course, of low cost. The nature of IoT devices and the huge amount of data generated by them causes serious failures in conventional network protocols, such as the traditional host-centered TCP/IP network: the address space is exhausted, and additional mobility-related mechanisms have been required. Moreover, the connection-oriented TCP requires resources (computing, memory and, energy) usually not available in small devices.

Information-Centric Networking (ICN) has been considered the most promising candidate to overcome the drawbacks of host-centric architectures when applied to IoT networks [3]. Conceptually, in ICN each piece of data has a unique, persistent and location-independent name that is directly used by

the applications for content search and retrieval. Therefore, ICN enables the deployment of in-network caching and content replication thus facilitating the efficient and timely delivery of information. However, the best qualities of ICN architectures can become restrictions in the IoT universe in Smart Cities. The basic and efficient mechanics of ICN of only responding with content to the interests posted on the network, can be a major obstacle. For example, when we spread a large set (thousands) of environmental sensors throughout the city, we would have sensors that periodically make their measurements available, resulting in an avalanche of requests and therefore a massive network overhead. In addition, the efficient caching mechanism would spread thousands of copies of interests/data whose validity is restricted.

To address the requirements of IoT communication models under the scope of ICN concept we are presenting ndnIoT-FC, a Named Data Network-based (NDN) architecture that respects the ICN rules but offers special treatment for IoT traffic. Our architecture combines efficient hybrid naming with strategies to minimize the number of Interests, applies publish-subscribe mechanisms and uses caching strategies that virtually eliminate copies of IoT data from the intermediate nodes. The ndnIoT-FC makes available new NDN-based application-to-application protocol to implement a signature model operation and tools to manage its life cycle.

The proposed strategies are efficient and lightweight and can be applied to devices with low capacity of computational resources and memory, being thus perfectly suited to operate on edge devices and take advantage of fog computing paradigm [4,5]. Our architecture also maintains total independence between network nodes, where decisions are made by the node without information or interference of the other nodes. To demonstrate its efficiency, we discuss the results of a simulated distributed gathering information system that collects measurements from environmental sensors installed in Data Collection Units (DCUs) scattered throughout the city. The collection services installed in the DCUs periodically transmit information to an IoT server (also denoted as broker) using the NDN concept. In a nutshell, the contributions of this work can be summarized as follows:

- a hybrid naming with strategies to minimize the number of travelling Interests in IoT applications;
- a distinguished NDN caching strategy for IoT friendly communications;
- new NDN-based application-to-application protocol for IoT applications;
- a heterogeneous and diverse evaluation setup considering both traditional and pub-sub communication models; and
- a clear improvement in terms of network overhead and Pending Interest Table (PIT) entry registries, while keeping the consumer satisfaction ratio.

The remainder of this paper is organized as follows: in the next section we present a review of concepts and related works. Section 3 describes the proposed architecture and strategies for an efficient management of IoT data in ICN-based environments. Section 4 details the simulation testbed and data sensor gathering use cases. In Section 5 we discuss the results. Finally, the conclusions are presented in Section 6.

## 2. Related Work

The innovative concepts of ICNs, such as naming, named-based forwarding and in-network caching bring new benefits for content delivery but is not suitable for all types of network traffic. The basic ICN mode of operation, where one Interest is transmitted for each data packet requested, is not the best strategy for several communication models. Continuous flow traffic (media stream, sensor data gathering), disclosure of unsolicited information (alert message dissemination), and real time notifications present a strong tendency to cause an avalanche of Interest packets, and of course, the increase in the network overhead [6].

In IoT environments typical producer applications send small datasets of information each time. These datasets are consumed by some types of applications, such as dashboards in Cloud or alert systems. ICN natively support pull traffic (each Interest is served by a data package), but is not appropriated for data pushing. To address the requirements of these types of traffic some works proposed a publisher-subscriber model (pub-sub), where consumers can make *signatures* of services to receive data without sending interests periodically [3]. The work in [7] classifies this type of traffic in single-request/multiple-response, periodic delivery,  $n$  responses and conditional delivery. In single-request/multiple-response a subscriber sends one request asking for data which may comprise of multiple responses spread over time. In periodic delivery consumers send one request packet asking for periodic data identified by name after a specific time interval. In  $n$  responses, subscribers nodes send one request packet for a specific number of responses, and in conditional delivery subscribers send a request packet to receive data from a publisher only if certain conditions are met or events triggered. Most of the work that implements pub-sub for ICN uses solutions based on *Persistent Interests* [8].

Moll et al. [9,10] improved and implemented the idea of Persistent Interests (PIs), and study their applicability in conversational services in NDN. They discussed the interplay of forwarding strategies and PIs in a way to improve the performance of the entire NDN. However, they compare the performance of PIs to the classical NDN approach using the network traffic generated by Internet telephony and do not discuss IoT environments.

In [6] two extensions to CCN's routing and forwarding for disseminating information represented as channels and real-time documents were presented. They combine PIs and Reliable Notifications to efficiently support these traffic types. Wang et al. [11] extended their previous work (COPPS [12]) with a lightweight implementation of pub-sub for IoT devices in NDN environments. The solution, called COPSS-lite, was developed to enhance CCN-lite [13] and also support multi-hop connection by incorporating the RPL protocol for low power and lossy networks [14]. However, such solution violates the loose coupling principle in their use of name-based routing or forwarding.

In [15], the authors discuss how NDN can support reliable push-based IoT traffic, through the definition of three strategies to ensure reliable data pushing between consumers and producers: Interest notification; unsolicited Data, and virtual Interest polling. They also propose a simple analytical framework that provides preliminary quantitative insights into the proposed solutions. However, the presented study, used as proof-of-content, focused on a single consumer-producer pair, at one-hop distance.

The work in [16] describes the potential of NDN in fog computing in smart environments, introducing a service orchestration mechanism targeted at a user-centric policy which values how the consumers perceive the services in terms of processing time as opposed to traffic in the edge domain. The solution proposed supports decisions of whether IoT data can be processed at the edge or remotely in cloud in a distributed manner. However, its focus is on the orchestration of services where the scale is much smaller than IoT environment where thousands of sensors are expected. This difference in scale is perceived in the proposed solution, which does not use pub-sub strategies to make ICN as a whole more efficient. Our solution aims precisely to meet this scale.

In [17] the authors discuss the impact that the receipt of considerable Interest packets by a specific NDN node can have in devices with limited PIT sizes and, present a fuzzy-based PIT-sharing algorithm addressed at solving these concerns. The work proposed makes an effort to identify an optimal node to serve as a sharing node, which can house the interest requests of several neighboring nodes and forward data accordingly, presenting improvements in account to both content delivery times and cache hit ratios, promoting the need for PIT size efficient solutions in NDN. However, the authors focus on a system supported by caching, not considering scenarios in which data is for immediate consumption, as is common in IoT environments.

In [18], the authors present a sensor as a service platform to host live content streams from a diverse set of devices. They propose a data dissemination layer that uses a publication/subscription overlay based on the ICN paradigm. They detail the communication between publishers and subscribers with the centralized broker and show two examples of data delivery on their platform, publishing data from drone events and publishing data from traffic sensors. However, the results are few, and there is no discussion about them. For example, they do not show how the number of Interest packets decreases. The work in [19] proposed HoP-and-Pull (HoPP), an interesting publish-subscribe scheme for typical IoT scenarios targeting IoT networks with hundreds of resource constrained devices at intermittent connectivity. The strategy limits the Forwarding Information Base (FIB) tables to a minimum and naturally supports mobility, temporary network partitioning, data aggregation and near real time reactivity. Zhang et al. [20] presented a prototype of an integrated framework, dubbed NDNoT, to support IoT over NDN and provides services such as auto configuration, service discovery, data-centric security, content delivery, and other needs of IoT application developers. However, in both works, the authors do not discuss the performance of the proposed solution.

The work in [21] proposed an IoT architecture and implementation details for devices and service networking, communication model, management, and naming. For each model they propose mechanisms supporting node mobility, hand-off, packet design, and push and pull data services without changing NDN data exchange model. The evaluation is done using ambient assisted living applications. The authors argue that their strategies based on semi-persistent interest, achieve lower control overhead and the losses related to mobility are kept at a minimum. However, a new table entity was introduced at each network node to handle subscription which essentially deviates from the native NDN design and can produce bottlenecks in memory limited devices. Finally, in [22] the authors discuss an IoT pub-sub architecture for 5G networks. They identify the main research challenges and possible solutions for scaling a pub-sub architecture to IoT applications on 5G networks. The main focus of the work is the distribution of brokers as a solution to address scale of this environment. They discuss a solution that use MQTT+ over an ICN-based architecture (POINT [23]). However, they do not present any implementation involving NDN or results about the proposed solution.

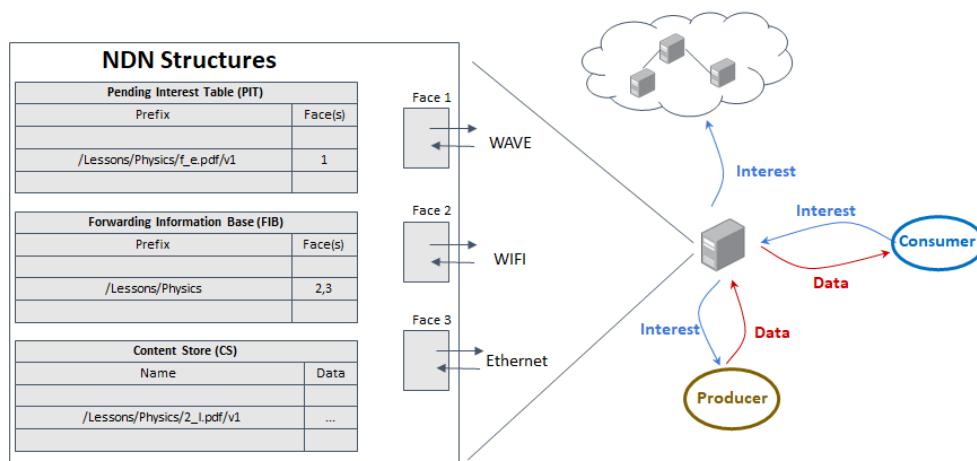
Our architecture improves the coupling between the ICN paradigm and IoT devices by making a better use of persistent Interests and a finer subscription management of IoT-related subscriptions, while focusing on the common IoT circumstance of data meant for immediate consumption where caching policies are found lacking. All the aforementioned characteristics were implemented without major changes to the core NDN entities, promoting side by side evaluations with the traditional NDN architecture and enabling the simultaneous use of NDN with pub-sub and non-pub-sub communication models, employing a challenging static scenario in which the absence of direct contact between publisher and subscriber nodes means managing of packets by intermediate nodes is decisive. The results, obtained considering distinct IoT traffic profiles, which is not common to see in the literature, will demonstrate the effectiveness of our solution namely in terms of network overhead and resource usage with respect to the number of PIT entries.

### 3. ICN Architecture for IoT Environments

Although ICN has been considered the most promising candidate for overcoming the disadvantages of host-centric architectures, the ICN architecture presents severe challenges in the IoT universe. To better explain how the solution proposed addresses these challenges, it is important to first describe the main characteristics of ICN and requirements of IoT environments.

### 3.1. ICN Basics

In Named Data Networking (NDN), an ICN-based architecture implementation, to receive Data a consumer sends out an Interest packet carrying a name that identifies the desired content. Routers remember the interface from which the request arrived, and then forwards the Interest packet by looking up the name in its Forwarding Information Base (FIB), which is populated by a name-based routing protocol. Once the Interest reaches a node that has the requested data, a Data packet is sent back in reverse path created by the Interest packet, back to the consumer. Routers store the Interest 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. The Data packets are stored in the Content Store (CS), which is basically the router's buffer memory subject to a cache replacement policy. Figure 1 details this process.



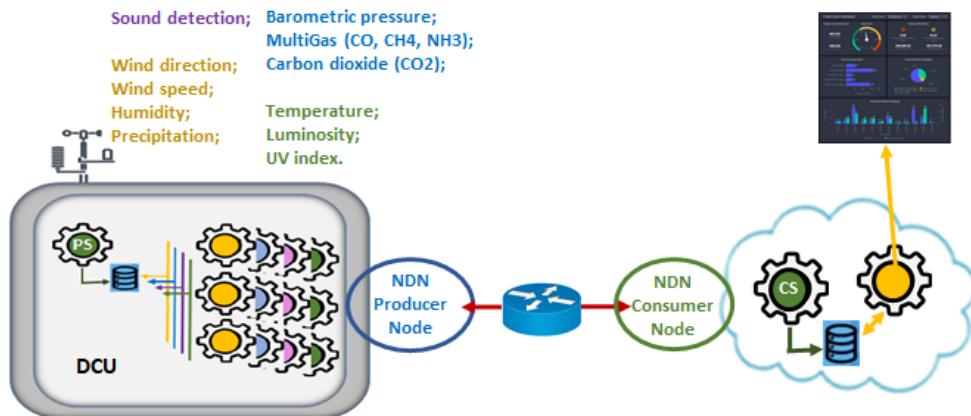
**Figure 1.** Packet forwarding in NDN.

### 3.2. IoT Characteristics

A typical IoT scenario for Smart Cities is shown in Figure 2. Sensor information collection units, also denoted as Data Collecting Units (DCUs), widely disseminated in the City, continuously collecting the environmental information to be transmitted to the Cloud. Each DCU is equipped with a large environmental monitoring sensor set that aims to, for example, collect relevant information about the environment condition in dense urban scenarios. The collected information are exposed through services and consumed by services installed in the Cloud [24], responsible for the organization of the information and making them available for user applications, such as dashboards.

Spread across the city, DCUs generate a brutal amount of information. Theoretically, within the ICN concept, each reading of each sensor should receive an Interest from each interested consumer, resulting in a flood of Interest packets that would inevitably cause the network overload. However, some features of the IoT environment can be useful when overcoming this challenge. A host can manage multiple DCUs, which in turn can contain multiple services that handle the various sets of sensors. In this way, it is possible to treat this whole set as a collection point and thus have a unique identification for the point (prefix name).

The collection of information from the sensors can be performed on demand for specific cases, but in general, it is done continuously. A subscription Interest system allows minimization of the amount of Interest packets on the network. However, such subscription system does not solve the caching flood caused by huge amount of Data packets. In general, the data from sensors are intended to Cloud-based applications, and for that reason they are not useful for others ICN nodes. Thus, the caching system must have a special role to treat data from IoT sensors.



**Figure 2.** IoT data gathering typical scenario.

It is expected that IoT applications can make optimizations, such as aggregating the collected data and transmitting them in a single data block using the network infrastructure more efficiently. However, to do so, they require a special service from the network. For IoT environments, it is important that the network provides a service for which applications can operate in publisher-subscriber (pub-sub) mode. A pub-sub service with simple interfaces allows not only standardization but also facilitates the interoperability between applications. In addition, optimizations at the network level improves the scalability of the entire infrastructure. The *ndnIoT-FC* architecture here proposed, presented in the next section, was designed to address these challenges.

### 3.3. *ndnIoT-FC* Architecture

The main focus behind our architecture is to address IoT services by means of extending the NDN concept, introducing publisher-subscriber mechanisms allowing for a reduction to in-network traffic, and implementing changes at the NDN forwarding module level. This provides versatility on how data packets are managed to better accommodate such communication model. To this extent, various characteristics inherent to NDN are crucial and need to be taken into account.

Packet naming, as one of the most important factors in Named Data Networks, must allow applications to subscribe to specific content without restricting how data should be forwarded.

In scenarios following a publisher-subscribe methodology data can be seen as extremely volatile, i.e., the lot of data periodically generated by IoT devices are for immediate consumption. This characteristic intensifying the need for Interest packets to be kept at a minimum, which in turn puts emphasis in how publisher nodes react to subscription requests and how frequent content should be updated.

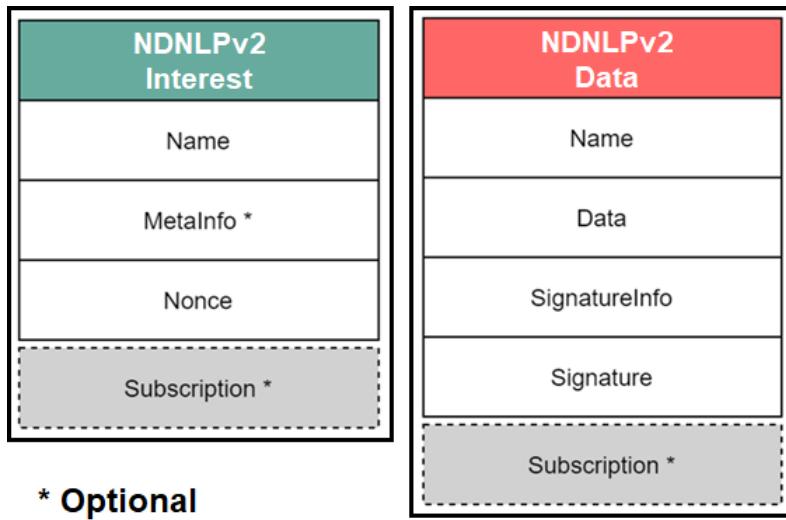
Likewise, this volatile nature of the data results in a high burden on the caching of intermediate nodes, as several updates for the same content are received and thus need to be maintained, which puts a lot of weight behind the design of efficient caching methods towards these occurrences, or can result in the disability of caching altogether.

In the forwarding spectrum, PIT entries validity and availability are major factors, as performance losses are evident in the event of lasting entries for nodes which are no longer available. Lastly, in the case of highly dynamic networks, as is the case of vehicular networks, paths should be updated frequently to ensure availability of content and allow efficient routing of data.

With the main objective of meeting the characteristics above described, we propose a set of mechanisms in the NDN base architecture that are described below.

### 3.4. ndnIoT-FC Pub-Sub Architecture

To maintain the low complexity of NDN communications, which makes use of two types of packets, Interest and Data packets, our focus was directed towards adjusting the current packet structure as opposed to creating an entirely new one. Thus, the addition of a new tag to the NDNLPv2 packet was needed. The new *Subscription* tag is used to differentiate native NDN communications from publisher-subscriber specific ones. The new packet structure is shown in Figure 3. Moreover, we employ the use of the well-known Named Link State Routing (NLSR) protocol [25] in order to hand out knowledge of which services are available in the network and, thus, allow nodes to generate content requests towards a specific DCU in the network. The protocol works by operating two parallel processes which make use of Interest/Data packet exchanges with a very specific packet name format introduced at the end of the packet name, allowing nodes to identify these packets as periodic routing table updates. First, neighboring nodes trade packets with their own router identification (e.g., /Network/Router\_ID/NLSR), to infer adjacency and identify available paths towards specific nodes in the network. Secondly, nodes which act as data producers send link state announcements (LSA), this is, packets with the name prefixes of the services they possess (e.g., /Router\_ID/Content\_Prefix/LSA) towards the network, advertising their availability to reply to requests. Based on the information resulting from these two processes, each node can build network topology of where and which content prefixes are available, information which is especially crucial for consumer nodes.



**Figure 3.** ndnIoT-FC pub-sub packet changes.

#### 3.4.1. Hybrid Naming Strategy

Under a Smart City context, sensory measurements are exposed by microservices hosted in the DCUs which can be managed by some remote host.

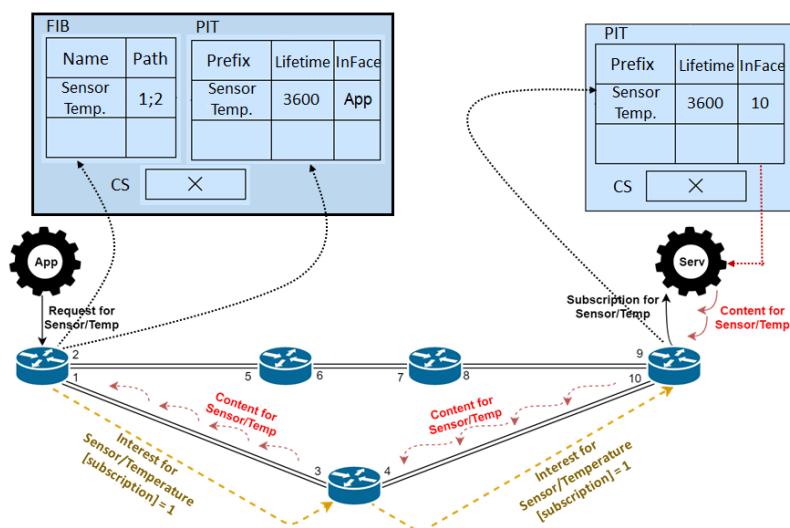
Whatever the configuration at the collection point, the subscription will be for a service that will periodically send a set of readings to the subscriber node, this is, the subscriber is not required to sign up for every single content available in each sensor or to a specific measurement. Moreover, the subscriber nodes possess brief knowledge of which services and DCUs exist in the network, which grants them the ability to construct an Interest packet appropriately named that can be used to retrieve the intended data. Accordingly, we adopted a hybrid naming strategy to ease the process of subscription of data available to the ICN subscriber nodes: Hosts, DCUs and services must have a static unique identification which can be combined in the following sequence: **host\_ID / DCU\_ID / service\_ID**, to form a specific

naming format which can handily be identified by intermediate nodes NDN tables and forwarded correspondingly. This naming format must be used between all the communications under the scope of the Publish–Subscriber mechanism, both by subscribers and publishers, unifying all content requests under the same naming format. As the use of a single subscription request allows for several data to be transmitted towards the subscriber for one specific service, the identification of the content itself coming from the sensors is at the discretion of the applications, without impacting the naming of the request. Additionally, it is important to notice that a single host may request content from services which can be spread across different DCUs. Figure 2 shows four groups of sensors exposed by four groups of services (yellow, blue, pink, and green). Although some services are available in different *localization*, i.e., are available by different Faces in terms of ICN architectures, it can also be induced that requests arrive at an intermediate node through the same incoming Face. In this way, we adopt the aggregation of names in the data structure (PIT) of the backbone nodes of our proposed architecture.

By using naming aggregation, we reduce PIT sizes as several entries for different contents can all be agglomerated, depending on which host originated the requests. The Interests arriving from the same subscriber node are registered in the PIT as a single entry, using the content name as a baseline, under the format **host\_ID** to establish the PIT entry name. This strategy adequately meets the scalability requirements of the IoT environment, where a large amount of microservices are needed to handle multiple sensors, as data must be forwarded towards the interested subscriber despite which sensor originated the content and intermediate nodes are only concerned in forwarding packets according to known paths. Thus, the allocation of resources towards PIT entry analysis in intermediate nodes is not linked to the number of services under subscription by the subscriber node. This format for naming aggregation is illustrated in the following section where we show an example of how the subscriber and intermediaries PITs are filled.

### 3.4.2. Forwarding Process

Although no changes were applied directly to the original NDN structures, the information they carry over the course of both Interest and Data packets, now is treated in a different manner. As illustrated in Figure 4, when subscription events are concerned, there are four key points to be addressed: *persistent interests*, *caching*, *entry management*, and *naming aggregation* explained in the previous section.



**Figure 4.** ndnIoT-FC pub-sub general communication.

**Persistent Interests:** PIT entries created as a result of an incoming Interest packet, marked as a subscription, are maintained over an extended period of time (*Lifetime* field in PIT Table), and are not erased after a successful content retrieval via a publisher node, allowing the same entry to be used for subsequent communications.

**Caching:** Data obtained from sensors in a Smart City are either used for aggregation systems for calculating average values, for joint analysis for wider observation or for real-time usage (indication of accidents, traffic lights, etc.). In all referenced cases they are for immediate consumption, i.e., they are periodic and temporary, a conduct that directly influences how caching strategies must react to deal with such volatile content, as caches need to be continuously updated for the most updated data, which puts a huge strain on already limited devices such as sensors, both from a caching size and energy-constrained perspective. To this extent, we argue that data resulting from subscription requests should not be added to the caching of intermediate ICN nodes, and in order to reduce the burden caused by continuously updated data deriving from the publisher-subscriber mechanism, while also guaranteeing its freshness, caching is disabled for such packets in the proposed solution.

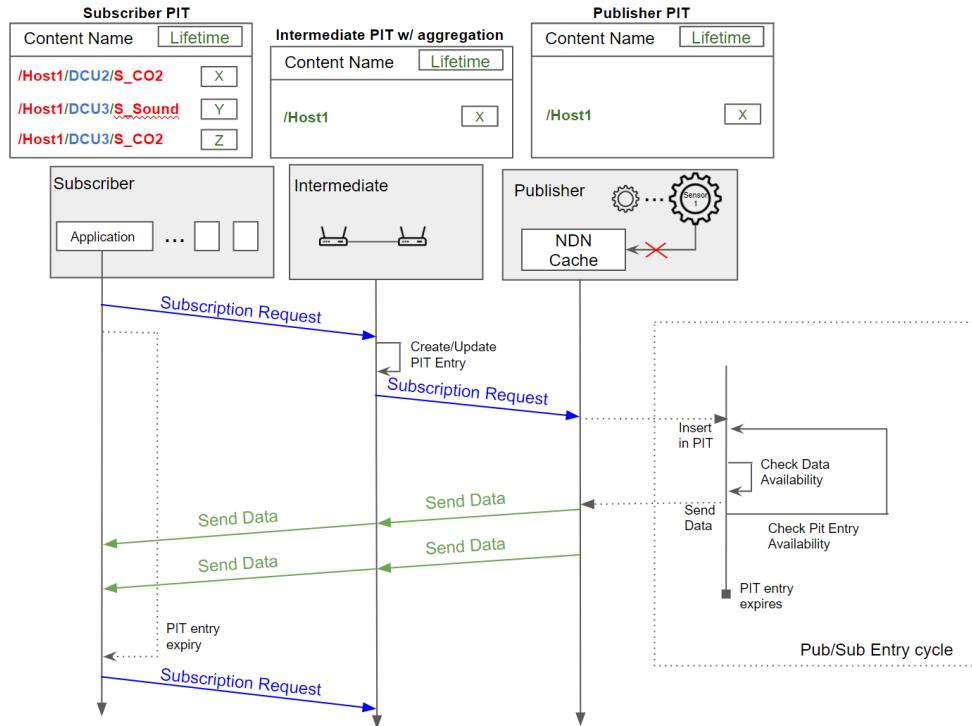
**Interest management:** when meddling with persistent PIT entries two key points need to be considered: entry lifetime, and path availability. On the first hand, since no acknowledgement packets are used in the NDN architecture, it is important to infer if a party invested in a specific subscribed content still exists and, therefore, if data should keep being transmitted towards known paths; On the other hand, in a dynamic network, paths need to be constantly updated in order to address availability concerns and to ensure data received can be successfully forwarded towards the end-user. To this extent, subscribers are required to periodically send new subscription packets for the same content to keep entries fresh on all nodes involved. When a PIT entry already exists for a received subscription packet, the *Lifetime* value of the received interest will be used as a replacement of the current PIT entry lifetime, serving as a token of the investment of the subscriber in continuing to receive the subscribed data. Furthermore, as we are dealing with a publish-subscribe mechanism where Interests are scarce compared to the native NDN workflow, this interest packet will be forwarded according to known paths in order to allow nodes further into the network to likewise update their own table entries, as opposed to the native mechanism where interests would be dropped.

As for the aggregated PIT entries, a single entry serves numerous incoming data packets, which means that only one lifetime value is attributed to the entry as a whole. In an effort to guarantee that all interests requests are offered a chance to be resolved, the lifetime value of the most recent Interest is used to update the aggregated PIT entry. This behavior allows intermediate nodes to keep forwarding content towards the subscriber until no remaining Interest exists in the services promoted by the specified DCU while also promoting cases where content keeps being delivered when the subscriber has not renewed investment in the data and PIT entries would have otherwise expired. This issue is handled by both the sensors installed in DCUs, which stop the Content delivery process as soon as the scheduled lifetime for their service expires, and subscriber/intermediate nodes which drop any non-intended Data packets received, an event easily identified as no aggregation occurs on their PIT tables.

Figure 5 shows the interaction of producers and consumers in our pub-sub architecture, whose conception was based on the following aspects:

- subscribers send requests for specific contents, as persistent Interests, with lifetime values (time interval between refresh of PIT entries);
- publisher nodes accept subscriptions for content they are producing, attending to them as soon as content is generated;
- publishers send data for subscribed content according to each individual request, in case the receiving paths differ, addressing multiple subscription requests for the same content via the PIT entries;

- management of subscription entries in the PIT is performed periodically, expiring after a certain period of time. A subscription might be refreshed if an Interest is received for a PIT entry already available;
- aggregation of PIT entries for services of same subscriber node in the intermediate nodes (red text in the Content name table of Figure 5), and
- treating all PIT entries as native, allowing publishers to update PIT entries to Persistent when data is delivered.



**Figure 5.** ndnIoT-FC pub-sub workflow.

### 3.4.3. Subscription Requests Workflow

As shown in Figure 6, at the beginning of communications a subscriber node, which is interested in subscribing a specific content, sends an NDN interest packet with a special “subscription” tag (mainly used to differentiate a native NDN content request from a publisher-subscriber node) towards its neighborhood according to existing FIB entries. Upon receiving this Interest packet, nodes intervening in the communication create a PIT entry as per usual NDN forwarding using an extended “lifetime” value defined by the subscriber node upon Interest creation. Then, likewise to the subscriber nodes, they forward the content to their known paths towards the publisher. When a subscription request reaches a publisher node, a PIT entry is created, as is dictated by the NDN stack and the underlying services installed at the node are informed of the occurrence, initiating a content creation and delivery process, marking the end of the subscription request.

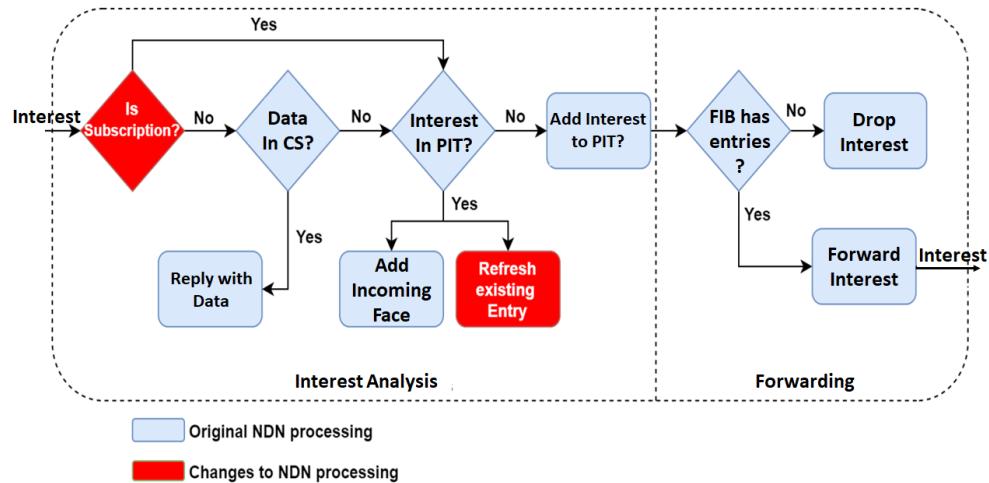


Figure 6. ndnIoT-FC pub-sub subscription request workflow.

When nearing the expiry of a PIT entry, as established by the lifetime value chosen during interest creation, if a subscriber still aims to maintain data receipts, it proceeds to send a new Interest packet for the same content, with the same Name, in order to relay this intent to the network. When this Interest is received by another node, if a PIT entry already exists for such content, the entry is updated with the received lifetime value, otherwise a new entry is created.

#### 3.4.4. Content Delivery workflow

As soon as the publisher service is notified of the Interest with respect to its content by another node, it verifies the available contents and generates a Data packet which will follow in-reverse path towards the subscriber node (Figure 7). However, contrary to the usual NDN forwarding mechanism, in which for any given Data packet there must be a corresponding Interest packet, the service will generate new packets periodically, after verifying that a PIT entry is still available for the content in question. This is made possible by managing the PIT entries existence and allowing them to be kept upon forwarding of a Data packet, meaning the path towards a subscriber node is always known by the Publisher until the entry expires. Thus, a new data packet will travel towards the subscriber periodically, containing updated content.

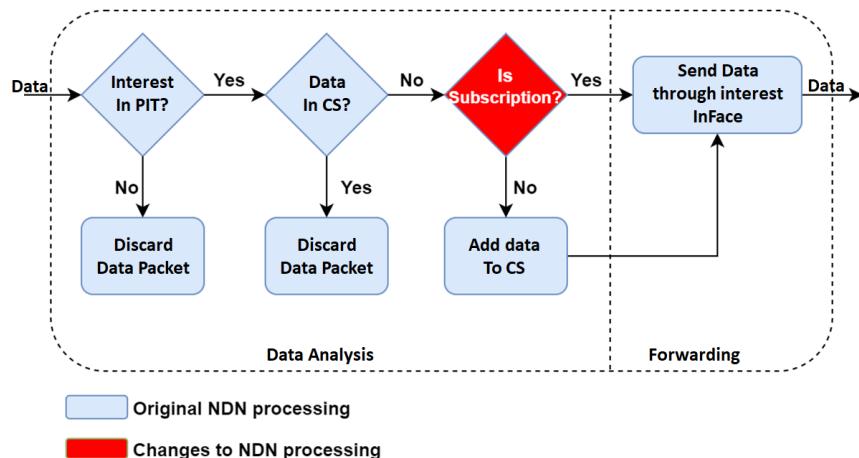


Figure 7. ndnIoT-FC pub-sub Content delivery workflow.

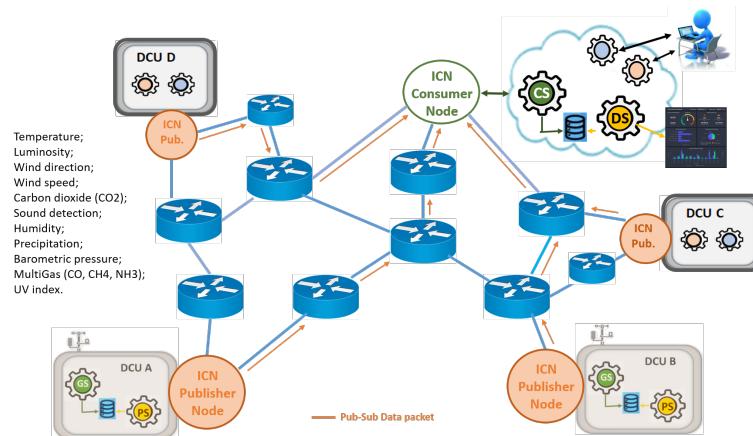
In short, the inserted strategies keep the characteristics of the NDN unchanged while incorporating publisher-subscriber mechanisms. As demonstrated by our results, our solution efficiently handles publisher-subscriber applications but remains equally efficient for traditional NDN applications.

#### 4. Testbed Platform and Use Cases

A common concern in Smart Cities is the focus on sensing procedures to provide city-wide information to city managers and citizens. To meet the growing demands of Smart Cities, the network must provide the ability to handle a large number of mobile sensors/devices, with high heterogeneity and unpredictable mobility, by collecting and delivering the sensed information for future treatment [24].

##### 4.1. Platform

In the scope of the IoT paradigm, the communication must allow the seamless integration of any object, allowing new forms of interaction between people and devices, or directly between devices (machine-to-machine). Our software testbed architecture, illustrated in Figure 8, aims to provide a city-wide scenario with heterogeneous elements. The platform can be divided into three main groups. Data producers (edge devices), data consumers (Cloud services and applications), and ICN routers (blue devices) to interconnect producers and consumers.



**Figure 8.** ICN IoT Platform with pub-sub services only.

On the edge of network we have the data producers, Data Collection Units (DCUs), where sensors sets (temperature, luminosity, sound, etc.) are housed (Such DCUs present a similar configuration as the ones placed in the city of Aveiro, under the scope of Aveiro STEAM City project (<https://uia-initiative.eu/en/uia-cities/aveiro>)). To control and capture the sensor measurements, the DCUs have a Gathering Service (denoted as GS) that periodically collects the data and stores it locally. The DCUs also contain the Producer Service (PS) that serves those interested in such information, i.e., the applications. All DCUs on our platform are configured as ICN nodes and prepared to send and receive packets according to this paradigm. As illustrated in Figure 8, for the use cases to be discussed in this paper, the DCUs will be ICN producers, as they are the source of sensor measurements.

The Cloud hosts the services and applications that subscribe the edge services to get the sensors information. As show in the Figure 8, we have a Consumer Service (CS) that periodically receives data from GSs and puts them into Cloud databases. Such information can be used for Cloud applications such as user dashboards. In addition, of course, the Cloud has at least one ICN node configured, the ICN consumer node.

As we will demonstrate in our use cases, such platform is flexible, scalable and can operate in a homogeneous or hybrid way. We can easily add services both in the Cloud and on the edge, we can configure routes with different link costs, can configure several types of communication interfaces operating simultaneously on the network nodes and also increase/decrease the number of sensors and the number of DCUs.

#### 4.2. Use Cases

To demonstrate the potential of our solution, we discuss the results of a simulated distributed gathering information system that collect measurements from environmental sensors installed in DCUs scattered throughout the city. The main idea is to explore the use of collection services installed in the DCUs to periodically transmit information to the Cloud services using the proposed pub-sub ICN solution. We compare such solution with the ICN native implementation where one Data packet is transmitted after the reception of an Interest packet. Both strategies will make use of the exact same content names, using the previously mentioned format, in which each Host, DCU and Service is considered unique in the context of the network and, thus, possesses a unique identifier, as the differences lie in the way packets are handled and forwarded and not how they are identified. We discuss our results using three distinct use cases, exploring different combinations and evaluating the scalability of ICN solution in the presence of different communication paradigms.

The first use case is quite simple. One consumer service hosted in the Cloud subscribes the data from one producer service installed in one DCU at the edge of network, i.e., the networks assumes only a pub-sub communication model. In the second use case we consider a network with distinct communication models, pub-sub and native communications. However, each producer node (DCU) attends to a single communication model. In the third use case we assume that both communication models can be assured by a single producer node. For that we use three types of services hosted by DCU. One of them, the PS, provides sensor information periodically following subscriptions made by the consumer nodes. The remaining two are an on-demand service used to provide customized amount of sensor data (RQ service) and for DCU management (MN service). From use case to use case we have progressively increased the number of services at the edge and in the Cloud to see how our strategies address network challenges in terms of network overhead, delivery ratio and resource usage.

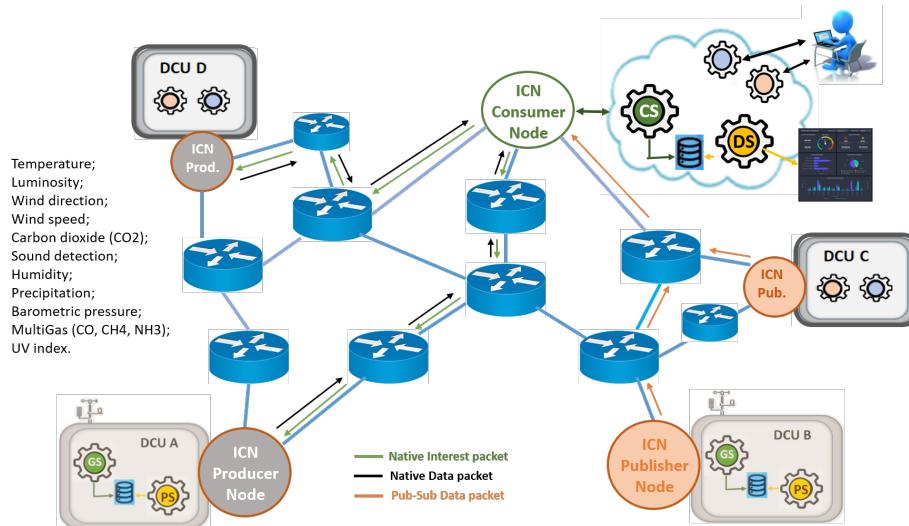
##### 4.2.1. One Consumer Service in the Cloud and One Producer Service at the Edge

The purpose of the first use case is to demonstrate the focus of our strategies to meet the requirements of applications where the premise of an Interest for each content is not performing well. As illustrated in Figure 8, a producer service (orange circle) receives “subscriptions” from a Consumer Service (CS in Cloud) and henceforth transmits data packets periodically (orange arrows). With this simple application we show how our network services for subscriptions facilitate the development of applications without distorting the basics of the ICN paradigm. We also demonstrate how pub-sub strategies can alleviate the Interest storm, optimize the ICN network caching while choosing the most efficient routes between producer and consumer.

##### 4.2.2. Different Types of Traffic in the Same ICN Network

In the second use case, we evaluate how our solution behaves when two types of applications with different requirements in terms of types of traffic (pub-sub and native ICN) coexist in the same ICN network. Nevertheless, we assume that one edge ICN node can only attend to a single traffic type (pub-sub or native ICN). To do that, we use the three types of services previously presented, where the subscription-type service is hosted by half of the DCUs (orange circles) and the native ICN services

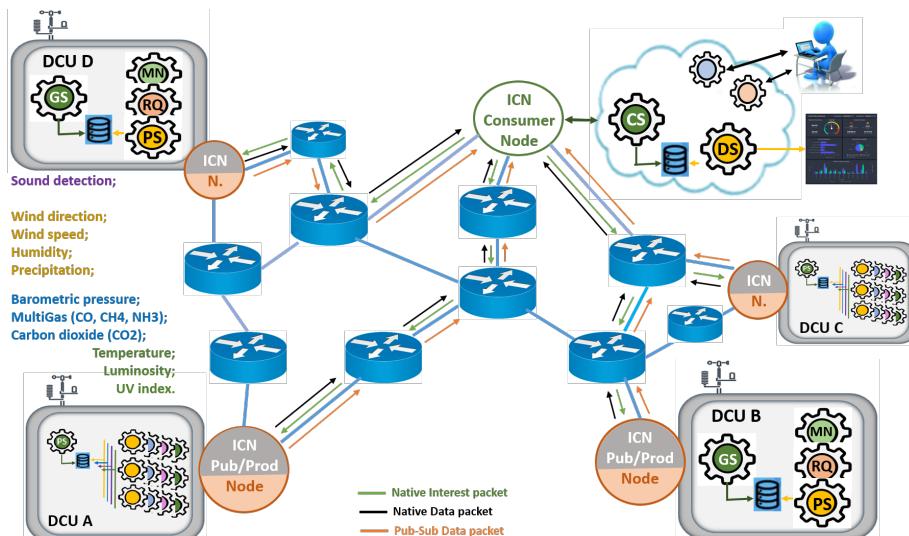
(RQ and MN, grey circles) are hosted by the other half, as illustrated in Figure 9. Native mechanism communications exchange a pair of Interest (green arrows) and Data packets (black arrows) whereas pub-sub makes use of several periodically transmitted Data packets.



**Figure 9.** Use case 2: multiples traffic types in the ICN.

#### 4.2.3. A Heterogeneous ICN with a Multitude of Traffic Types

The objective of this third use case is to evaluate the scalability and elasticity of the entire network where the same ICN node implements both types of traffic (pub-sub and native ICN). In contrast to the previous use case, the ICN producer node will have to handle subscriptions from the ICN consumer together with requests based on Interest packets, following the traditional ICN paradigm (dual colored circles). As illustrated in Figure 10, we have progressively increased the number of services at the edge and in the Cloud.



**Figure 10.** Use case 3: multiples services in the same ICN node.

## 5. Performance Evaluation

To evaluate the proposed publisher-subscribe mechanism, we will focus on the burden caused by network overhead according to the different use cases previously described, in which both Data providers and content subscribers are non-mobile and are connected through various intermediate nodes. The simulations, carried out using the NDNsim framework, were performed throughout a five minutes period, during which subscribers would notify the network of their interest in a certain content via an Interest packet, which remains in PIT tables for 15 seconds, time after which the entry will expire (subscriber methodology). In case a subscriber still requires the content after this period, a new Interest is issued to update the existing PIT entries. On the other hand, publishers will react to incoming Interest events, by sending content every two s. These parameters are illustrated in Table 1, as well as the network topology considered, depicted in Figures 8–10.

**Table 1.** ICN node simulation parameters.

Subscribers		Publishers	
Entry Lifetime (LifeT)	15 s	Data Generation	On Wake Up
Interest Retransmission	LifeT-1.0sec	Transmission Interval	2 s
Cache Size	0	Cache Size	0
Installed Technology	Ethernet	Installed Technology	Ethernet
Mobility	Fixed	Mobility	Fixed

The number of packets counted in each use case is retrieved by registering every occurrence of a received/sent Interest/Data packet during simulation for each of the 15 nodes considered and performing a sum of the values of all nodes registered according to their type (e.g., Sent Data packets). This allows us to understand metrics such as the network overhead and how much unnecessary traffic is being propagated throughout the network. Furthermore, as no acknowledgements exist, we can calculate how many Interests are generated by each subscriber node, by knowing the total simulation time, which was 5 min, and the rate of Interest retransmissions. For the native case, every Interest transmission was set at 2 s, which means that every consumer node will generate 150 Interests to the network for each subscribed service; in the pub-sub case, retransmission of persistent interests was set at 14 s, meaning that every subscriber node will renew each content 21 times, thus, generating 22 packets towards the network for a given service.

A comparison will be made between the native NDN forwarding, in which one Data packet refers to an Interest, and the publisher-subscriber mechanism described in Section 3, given that caching will need to be disabled, as we aim to evaluate scenarios in which data is volatile, and therefore should not be stored locally. Lastly, we study the impact aggregation can have in PIT table sizes, and how they can improve performance.

Table 2 describes the number of nodes and their types across all the Use Cases, from a global count of 15 nodes comprising the network topology. In both variants of Use Case 1, the evaluation of each strategy was done separately, so it makes sense that same nodes were selected across both simulations, in order to guarantee that results obtained are valid and able to be compared, the difference being that in the first variant only a single node is considered as publisher, whereas the second scenario was scaled to have three different publishers. The remaining Use Cases, 2 and 3, refer to a scenario in which both the native NDN and proposed pub-sub were coexisting. In the former, each of the two available applications were installed in a pair of nodes, meaning that four different entities will be servicing either pub-sub or native requests at any given time, with the remaining ten nodes acting as intermediate and forwarding packets between each end-point. As for the latter, each of the four previously selected content delivery nodes will act in behalf of both strategies, which means they will satisfy both native NDN and pub-sub services

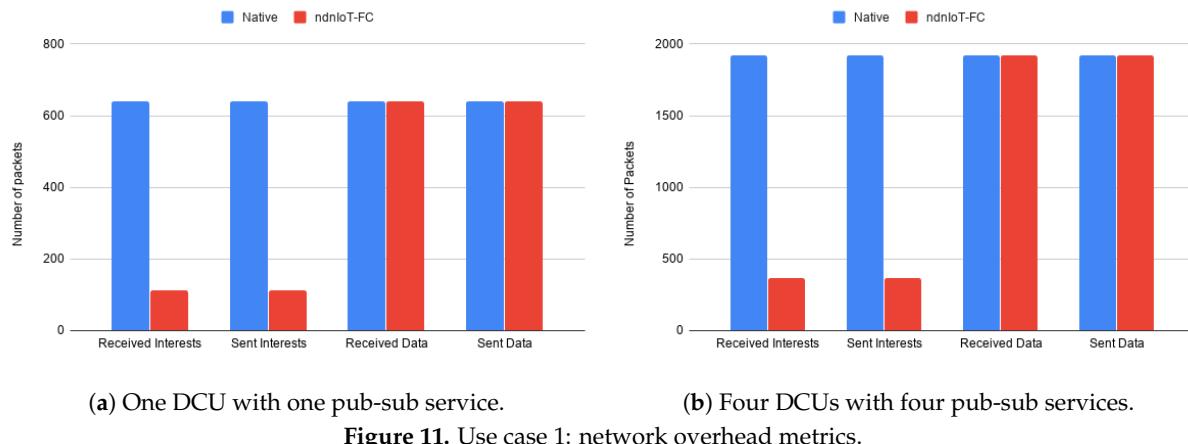
according to requests received, in counts of 2 and 3 services installed respectively, while maintaining the same number of intermediate nodes.

**Table 2.** Network topology—Node count per Use Case.

Subscriber/ Producer	Publishers	Producers	Intermediate	Global Node Count
Use Case 1.1		1 (same node for both strategies)	13	
Use Case 1.2		3 (same nodes for both strategies)	11	
Use Case 2	1	2 (1 service each)	2 (1 service each)	10
Use Case 3		4 (3 pub-sub and 2 native services each)	10	15

### 5.1. Use Case 1—An ICN with Pub-Sub Only

The first use case refers to a small scenario in which an application, a Cloud service, a single subscriber to a single service installed on a single DCU. Results on the amount of sent and received Interests and Data packets are illustrated in Figure 11a, for both pub-sub and native NDN approach. The results show that both strategies were able to deliver the same amount of information, namely Data packets, with the pub-sub mechanism presenting a significant decrease in the number of transmitted packets required to search for the content, as the result of the pub-sub mechanism.



**Figure 11.** Use case 1: network overhead metrics.

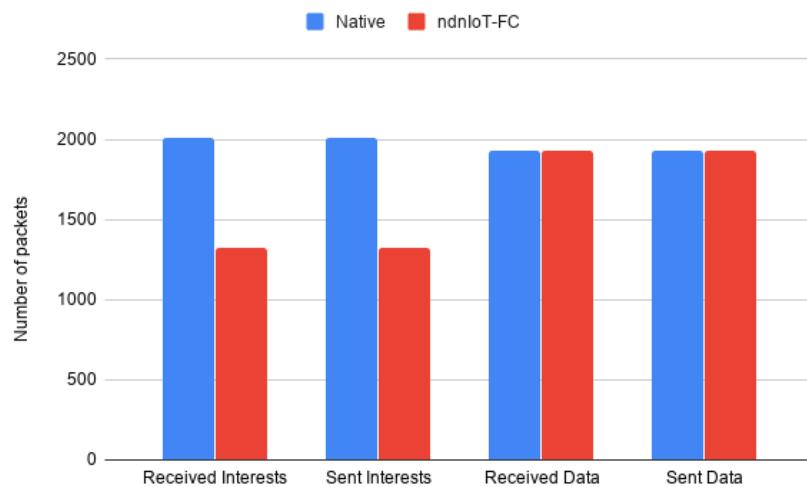
The previous scenario was extended to include three additional ICN nodes (DCUs), each one hosting a different pub-sub service subscribed by the same ICN node, the ICN consumer node (cloud node). Similar behavior to the previous case was experienced. As depicted in Figure 11b, despite the expected increase in the number of Interest packets, the ratio between the base version and our pub-sub implementation remains. The pub-sub version maintains the efficient delivery of Data packets using only 18% of the Interest packets observed in the native version.

### 5.2. Use Case 2—Different Types of Traffic

The second use case evaluated the performance of our solution by introducing different types of applications with different requirements in relation to the management of packages in the network. To do that, instead of having only pub-sub communication profiles, we mixed up pub-sub services with native

NDN services, i.e., those requiring an Interest packet for the resolution of a Data packet, across two of the four DCUs considered, while maintaining the number of subscribers at a single node. Thus, the scenario consisted of two different publisher-subscribe services made available to the subscriber, each one on single ICN node (DCU), and two native NDN services, for the remaining ICN nodes, in which Interests are not persistent. Furthermore, the publisher-subscribe communications, in this specific use case, require a higher number of hops towards the content, which directly impacts the number of packets registered.

The results illustrated in Figure 12 show that not only both services can work concurrently, but a reduction in the overall network traffic is noticeable, when compared to a pure native NDN communication model, since some of the content can be delivered by using the same PIT entries created upon interest receipt instead of recurring to new ones.



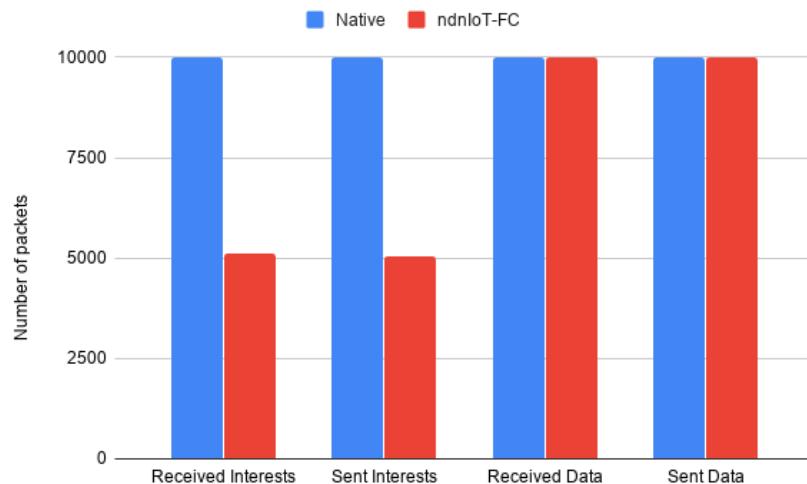
**Figure 12.** Use case 2—Network overhead metrics.

Despite the topology in this use case being equivalent to the second scenario of the previous use case, by switching two native NDN services by pub-sub services, a linear reduction in the number of required Interest packets is not observed as would be expected. This occurs due to the distance in which those services were installed (right side of Figure 8, as the native NDN communications will require less hops than their counterpart, thus provoking less ambiguous traffic in the network. This use case ultimately proves that both types of services can coexist and do not affect each other.

### 5.3. Use Case 3—Heterogeneity and Scalability

In the third use case, we are interested in the behavior of our solution as new services come on stream, i.e., the scalability of the entire network. The previous cases reveal that while a pub-sub method is ideal for volatile data delivery, it does not interfere with data using the native NDN forwarding mechanism, should it be necessary. To further test scalability, we increase the number and type of services running on each DCU depicted in Figure 10, three pub-sub and two native types, coming to a total of 20 services able to be subscribed to by the Cloud application. This behavior will result in an upsurge of information from several different types of applications coming from the same publisher node, which will be reflected on PIT entries of intermediate nodes, as well as a significant increase of both interest and data traffic in the network towards the same subscriber node, in order to further examine how the system behaves under heavier load.

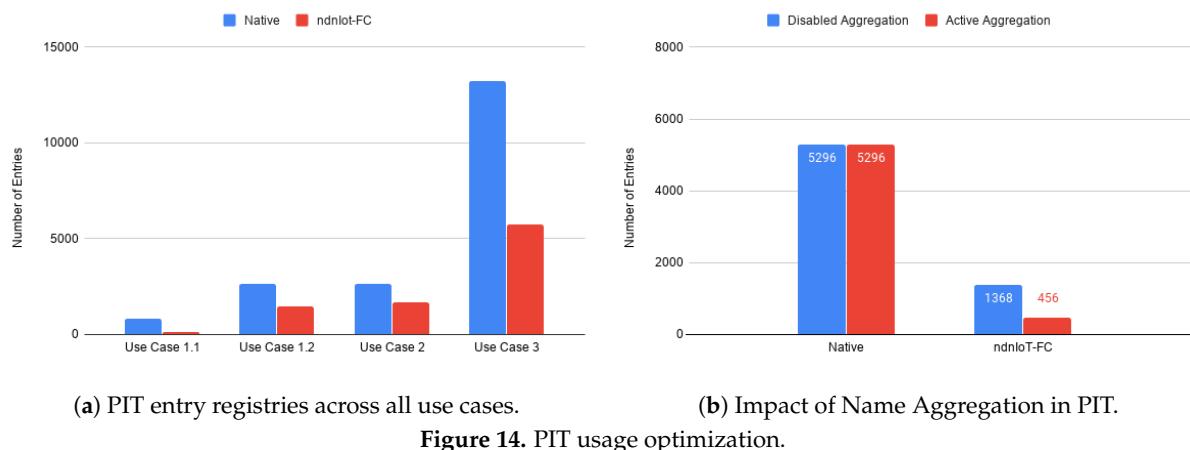
Results illustrated in Figure 13 show that by using pub-sub services to deliver volatile data, we were able to reduce Interests by half of the total needed by the native approach, resulting in a significant decrease in-network traffic, and thus, overhead. Additionally, when reducing the number of packets in the network we are reducing the energy consumption associated with the transmission and packet processing [26].



**Figure 13.** Use case 3—Network overhead metrics.

#### 5.4. PIT Size

One of the main topics of discussion about NDN performance is the number of entries kept at the PIT, which scales directly with network traffic leading to congestion situations. By introducing aggregation to our pub-sub solution, when services of the same DCU are concerned, we aimed to reduce the number of entries necessary to be kept at PIT tables. Proving the efficiency of this PIT management, the results in Figure 14a exhibit a strong reduction in the number of PIT entries across all use cases.



**Figure 14.** PIT usage optimization.

To understand how efficient Name aggregation can be in the pub-sub mechanism considered, we can look further into the results of ndnIoT-FC in use case 3, with emphasis on the number of PIT entries that were newly created or updated throughout the simulation, as depicted in Figure 14b.

Considering that no changes were made to the native forwarding mechanism in the context of this work, it is expected that no improvements are noticeable. In ndnIoT-FC, however, by using part of the

interest packet Names, which may include information about the kind of service under subscription, as opposed to the full name in order to construct a PIT entry, we were able to reduce the number of PIT entries required by one third using name aggregation, as services from the same location can be grouped into one single entry. This is especially meaningful when considering that an increased number of services made available will result in a higher number of PIT entries, which can be better controlled by the solution proposed. Additionally, the number of PIT entries reduces the energy consumption on each node by reducing the lookup stage searching for the pretending entry [27].

## 6. Conclusions

In this paper, we present and evaluate an NDN architecture following a publisher-subscribe mechanism, addressing different types of IoT traffic, which makes use of both packet Naming and clever PIT entry management to reduce network traffic. On one hand, it makes an effort to describe how publisher-subscribe solutions can greatly improve IoT traffic treatment in NDN. On the other hand, it evaluates how Naming aggregation can reduce PIT entry sizes, thus targeting network performance by means of faster look-ups. The results obtained show a clear improvement in terms of network overhead across several use cases when compared to the existing NDN Least-Cost approach. Additionally, the pub-sub solution can vastly reduce the number of PIT entries registered throughout the simulation, without impacting data delivery.

For future work we will focus on improving the overhead introduced by both subscription requests using the wireless medium and retransmission of those requests when PIT entries expire, resulting in a burst of information on intermediate nodes. Another aspect to be improved in future works is the management of subscription in mobile scenarios. When in dynamic topologies, as paths are never updated until the next PIT entry is refreshed, there is a chance that some entries might become outdated given the new location of Publishers or Consumers. Such behavior may lead to broken paths, and consequently, to unsatisfied Consumers.

**Author Contributions:** All authors designed the solution, analyzed the results and wrote the paper. L.G. implemented the solution, prepared the evaluation scenarios and the results. C.S. and M.L. supervised the entire research process. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work is supported by the European Regional Development Fund (FEDER), through the Competitiveness and Internationalization Operational Programme (COMPETE 2020) of the Portugal 2020, Regional Operational Program of Lisbon (FEDER); and Foundation for Science and Technology, project InfoCent-IoT (POCI-01-0145-FEDER-030433).

**Conflicts of Interest:** The authors declare no conflict of interests.

## References

1. Zanella, A.; Bui, N.; Castellani, A.; Vangelista, L.; Zorzi, M. Internet of Things for Smart Cities. *IEEE Internet Things J.* **2014**, *1*, 22–32. [[CrossRef](#)]
2. Vermesan, O.; J. Bacquet, E. *Next Generation Internet of Things, Distributed Intelligence at the Edge and Human Machine-to-Machine Cooperation*; River Publishers: Gistrup, Denmark, 2018.
3. Shang, W.; Bannis, A.; Liang, T.; Wang, Z.; Yu, Y.; Afanasyev, A.; Thompson, J.; Burke, J.; Zhang, B.; Zhang, L. Named Data Networking of Things (Invited Paper). In Proceedings of the 2016 IEEE First International Conference on Internet-of-Things Design and Implementation (IoTDI), Berlin, Germany, 4–8 April 2016; pp. 117–128. [[CrossRef](#)]
4. Zhao, Y.; Wang, W.; Li, Y.; Colman Meixner, C.; Tornatore, M.; Zhang, J. Edge Computing and Networking: A Survey on Infrastructures and Applications. *IEEE Access* **2019**, *7*, 101213–101230. [[CrossRef](#)]
5. Ullah, R.; Ahmed, S.H.; Kim, B. Information-Centric Networking With Edge Computing for IoT: Research Challenges and Future Directions. *IEEE Access* **2018**, *6*, 73465–73488. [[CrossRef](#)]

6. Tsilopoulos, C.; Xylomenos, G. Supporting Diverse Traffic Types in Information Centric Networks. In Proceedings of the ACM SIGCOMM Workshop on Information-Centric Networking, ICN’11, Dirk Trossen, UK, 19 August 2011; Association for Computing Machinery: New York, NY, USA, 2011; pp. 13–18. [[CrossRef](#)]
7. Nour, B.; Sharif, K.; Li, F.; Yang, S.; Mounbla, H.; Wang, Y. ICN Publisher-Subscriber Models: Challenges and Group-based Communication. *IEEE Netw.* **2019**, *33*, 1–8. [[CrossRef](#)]
8. Yao, C.; Fan, L.; Yan, Z.; Xiang, Y. *Long-Term Interest for Realtime Applications in the Named Data Network*; AsiaFl 2012 Summer School: Kyoto, Japan, 2012.
9. Moll, P.; Janda, J.; Hellwagner, H. Adaptive Forwarding of Persistent Interests in Named Data Networking. In Proceedings of the 4th ACM Conference on Information-Centric Networking, ICN’17, Berlin Germany, 26–28 September 2017; Association for Computing Machinery: New York, NY, USA, 2017; pp. 180–181. [[CrossRef](#)]
10. Moll, P.; Theuermann, S.; Hellwagner, H. Persistent Interests in Named Data Networking. In Proceedings of the 2018 IEEE 87th Vehicular Technology Conference (VTC Spring), Porto, Portugal, 3–6 June 2018; pp. 1–5. [[CrossRef](#)]
11. Wang, H.; Adhatarao, S.; Arumaithurai, M.; Fu, X. COPSS-lite: Lightweight ICN Based Pub/Sub for IoT Environments, 2017. Available online: <https://arxiv.org/abs/1706.03695> (accessed on 21 November 2020).
12. Chen, J.; Arumaithurai, M.; Jiao, L.; Fu, X.; Ramakrishnan, K.K. COPSS: An Efficient Content Oriented Publish/Subscribe System. In Proceedings of the 2011 ACM/IEEE Seventh Symposium on Architectures for Networking and Communications Systems, Brooklyn, NY, USA, 3–4 October 2011; pp. 99–110. [[CrossRef](#)]
13. Jacobson, V.; Smetters, D.K.; Thornton, J.D.; Plass, M.F.; Briggs, N.H.; Braynard, R.L. Networking Named Content. In Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies, CoNEXT’09, Rome, Italy, 1–4 December 2009; ACM: New York, NY, USA, 2009; pp. 1–12. [[CrossRef](#)]
14. Sobral, J.; Rodrigues, J.; Rabélo, R.; Al-Muhtadi, J.; Korotaev, V. Routing Protocols for Low Power and Lossy Networks in Internet of Things Applications. *Sensors* **2019**, *19*, 2144. [[CrossRef](#)] [[PubMed](#)]
15. Amadeo, M.; Campolo, C.; Molinaro, A. Internet of Things via Named Data Networking: The support of push traffic. In Proceedings of the 2014 International Conference and Workshop on the Network of the Future (NOF), Paris, France, 3–5 December 2014; pp. 1–5. [[CrossRef](#)]
16. Amadeo, M.; Ruggeri, G.; Campolo, C.; Molinaro, A.; Loscri, V.; Calafate, C.T. Fog Computing in IoT Smart Environments via Named Data Networking: A Study on Service Orchestration Mechanisms. *Future Internet* **2019**, *11*, 222. [[CrossRef](#)]
17. Maheswari, P.; Manickam, P.; Kumar, K.; Maseleno, A.; Shankar, D. Bat optimization algorithm with fuzzy based PIT sharing (BF-PIT) algorithm for Named Data Networking (NDN). *J. Intell. Fuzzy Syst.* **2019**, *37*, 1–8. [[CrossRef](#)]
18. Shariat, A.; Tizghadam, A.; Leon-Garcia, A. An ICN-based publish-subscribe platform to deliver UAV service in smart cities. In Proceedings of the 2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), San Francisco, CA, USA, 10–14 April 2016; pp. 698–703. [[CrossRef](#)]
19. Gündoğran, C.; Kietzmann, P.; Schmidt, T.C.; Wählisch, M. HoPP: Publish-subscribe for the Constrained IoT. In Proceedings of the 5th ACM Conference on Information-Centric Networking, ICN’18, Boston, MA, USA, 21–23 September 2018; ACM: New York, NY, USA, 2018; pp. 216–217. [[CrossRef](#)]
20. Zhang, Z.; Lu, E.; Li, Y.; Zhang, L.; Yu, T.; Pesavento, D.; Shi, J.; Benmohamed, L. NDNoT: A Framework for Named Data Network of Things. In Proceedings of the 5th ACM Conference on Information-Centric Networking, ICN’18, Boston, MA, USA, 21–23 September 2018; ACM: New York, NY, USA, 2018; pp. 200–201. [[CrossRef](#)]
21. Nour, B.; Sharif, K.; Li, F.; Mounbla, H. A Distributed ICN-Based IoT Network Architecture: An Ambient Assisted Living Application Case Study. In Proceedings of the GLOBECOM 2017—2017 IEEE Global Communications Conference, Singapore, Singapore, 4–8 December 2017; pp. 1–6. [[CrossRef](#)]
22. Redondi, A.E.C.; Arcia-Moret, A.; Manzoni, P. Towards a Scaled IoT Pub/Sub Architecture for 5G Networks: The Case of Multiaccess Edge Computing. In Proceedings of the 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), Limerick, Ireland, 15–18 April 2019; pp. 436–441. [[CrossRef](#)]

23. Trossen, D.; Reed, M.J.; Riihijärvi, J.; Georgiades, M.; Fotiou, N.; Xylomenos, G. IP over ICN—The better IP? In Proceedings of the 2015 European Conference on Networks and Communications (EuCNC), Paris, France, 29 June–2 July 2015; pp. 413–417. [[CrossRef](#)]
24. Almeida, R.; Oliveira, R.; Luís, M.; Senna, C.; Sargent, S. A Multi-Technology Communication Platform for Urban Mobile Sensing. *Sensors* **2018**, *18*, 1184. [[CrossRef](#)] [[PubMed](#)]
25. Wang, L.; Lehman, V.; Mahmudul Hoque, A. K. M.; Zhang, B.; Yu, Y.; Zhang, L. A Secure Link State Routing Protocol for NDN. *IEEE Access* **2018**, *6*, 10470–10482. [[CrossRef](#)]
26. Shen, Q.; Wu, J.; Li, J. Edge Learning based Green Content Distribution for Information-Centric Internet of Things. In Proceedings of the 2019 42nd International Conference on Telecommunications and Signal Processing (TSP), Budapest, Hungary, 1–3 July 2019; pp. 67–70.
27. Zheng, X.; Wang, G.; Zhao, Q. A Cache Placement Strategy with Energy Consumption Optimization in Information-Centric Networking. *Future Internet* **2019**, *11*, 64. [[CrossRef](#)]

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).