Multi-Source Data Retrieval in IoT via Named Data Networking

Marica Amadeo
University "Mediterranea"
DIIES Department
Reggio Calabria, Italy
marica.amadeo@unirc.it

Claudia Campolo
University "Mediterranea"
DIIES Department
Reggio Calabria, Italy
claudia.campolo@unirc.it

Antonella Molinaro
University "Mediterranea"
DIIES Department
Reggio Calabria, Italy
antonella.molinaro@unirc.it

ABSTRACT

The new era of *Internet of Things* (IoT) is driving the revolution in computing and communication technologies spanning every aspect of our lives. Thanks to its innovative concepts, such as named content, name-based routing and in-network caching, *Named Data Networking* (NDN) appears as a key enabling paradigm for IoT. Despite its potential, the support of IoT applications often requires some modifications in the NDN engine for a more efficient and effective exchange of packets.

In this paper, we propose a baseline NDN framework for the support of *reliable* retrieval of data *from different wireless producers* which can answer to the *same Interest* packet (e.g., a monitoring application collecting environmental data from sensors in a target area). The solution is evaluated through simulations in ndnSIM and achieved results show that, by leveraging the concept of EXCLUDE field and ad hoc defined schemes for Data suppression and collision avoidance, it leads to improved performance in terms of data collection time and network overhead.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless communication

General Terms

Design; Performance

Keywords

Named Data Networking; Internet of Things; data retrieval; naming; transport

1. INTRODUCTION

The advent of low-cost, low-profile electronic devices along with the advancement in communication technologies is enabling more objects around us to gain intelligence and ability to interact with one another, with the surrounding environment, with humans, and with remote systems via the Internet, hence bringing the Internet of Things (IoT) vision to life [1].

Thanks to its simple, connectionless communication model decoupling content from location, Named Data Networking (NDN) [2], one of the most noticeable information-centric networking architecture, appears as a promising solution for IoT [3]. Indeed, NDN systems give a *name* to the information instead of addressing a specific host/server, so consumers of data specify *what* they search for and not *where* they expect it to be provided. The consumer simply transmits an *Interest* packet to trigger the transmission of a *Data* packet from a potential producer; each received Data *consumes* the pending Interest in the forwarding nodes (*1-to-1* Interest and Data matching) so unsolicited and duplicated Data are avoided.

This shift from the IP-based host-centric paradigm of the traditional Internet to content-centric networking suits the IoT deployment (for example, many IoT applications request a *data item* regardless from the specific identifier of the producer; e.g., a temperature measurement in a given area), while promising simplicity and scalability in the application design, robustness and energy efficiency in massive data access [3].

A typical IoT traffic pattern that is not natively matched by the NDN paradigm is what we refer to as *multi-source data retrieval*, which is the focus of this paper. In this case, a consumer is interested in retrieving *different data items* of the same type from *different sources* at the same time. Data can be related to a particular event or collected from a given geographic area, with the consumer that is aware of or oblivious to the *number* and *identity of producers*. For instance, respectively, a monitoring system (consumer) asks for the energy consumption measurements of *all appliances* (producers) in a home; or a traffic control center is interested in retrieving accurate and reliable traffic information from vehicles in a specific road segment.

The *1-to-1* Interest-Data matching of vanilla NDN would require issuing an Interest packet to retrieve *each Data item*, and may be inefficient in terms of network and device resources. A more efficient solution would rather be the transmission of *a single Interest* to retrieve *many data items at once*. The requested changes to the NDN forwarding fabric would be small and the advantages manifold. First, *multi-source data retrieval* could leverage the expressiveness of the hierarchical Universal Resource Identifier (URI)-like NDN naming scheme to manage the multiple answers. Second, is-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
ICN'14, September 24–26, 2014, Paris, France.
Copyright 2014 ACM 978-1-4503-3206-4/14/09 ...$15.00.
http://dx.doi.org/10.1145/2660129.2660148 .

suing a single Interest would avoid the redundancy of Interest packets circulating into the network, hence saving bandwidth.

Despite the intuitive convenience of such a solution, releasing the *1-to-1* matching between Interest and Data, conceived to ensure flow control and reduce the amount of unwanted data, would pose some issues that could be exacerbated in a wireless environment. For example, multiple nodes that store or generate Data matching the Interest (providing the same type/piece of information) risk colliding when attempting to reply to the same Interest, with consequent Data packet losses that would undermine reliability.

The forwarding fabric and retransmission routines at the NDN *Strategy* layer need to be re-engineered to improve *diversity* and *reliability* in data retrieval while ensuring *efficiency*. In such a context, the contributions of the paper are as follows:

- we clearly identify some representative IoT use cases where NDN multi-source data retrieval can be beneficially exploited and we debate issues that may arise;
- we design the overhauling of the vanilla NDN forwarding engine in order to support IoT multi-source data retrieval, with focus on producers deployed in a wireless environment;
- we implement *(i) distributed consumer-aided collision avoidance* and *Data suppression* mechanisms, enforced at the producer-side, to improve content retrieval from multiple sources; and *(ii)* a *retransmission routine* to counteract Data losses that relies on the NDN built-in EXCLUDE field in Interest packets, adequately rethought to allow selective Data retransmissions.

The remainder of the paper is organized as follows.

Section 2 introduces the fundamentals of NDN and debates the potential of this paradigm in IoT environments. Section 3 discusses multi-source data retrieval and representative IoT use cases with relevant open design issues. Details of our proposal are provided in Section 4, while Section 5 analyses the simulation results. Conclusive remarks are wrapped up in Section 6.

2. NDN FOR IOT

2.1 NDN in a nutshell

The NDN architecture is based on the *Content Centric Networking* (CCN) proposal, presented by Van Jacobson *et al.* in [4]. Communication is based on hierarchical, application-specific content names and two packet types: the *Interest*, used to ask for a content, and the *Data*, used to answer the Interest by carrying the content itself. Content integrity and authenticity are ensured by piggybacking the data publisher's signature and other authentication information in each Data.

In addition to the name of the requested content, an Interest can include several optional fields, including the following ones, which are relevant in the context of this paper: *(i)* EXCLUDE, which specifies the components that should not appear in the name of content returned in response; *(ii)* ANSWERORIGINKIND, which specifies if the requested content could be obtained from intermediate node or must be generated by the producer; *(iii)* SCOPE, which limits the number of hops an Interest can propagate, e.g., Scope 2 limits its propagation to the first-hop node.

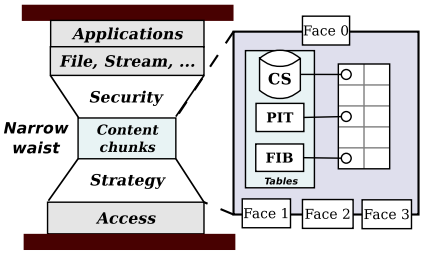


Figure 1: NDN hourglass model.

NDN also defines a specific node architecture and a set of basic packet processing rules.

Node model. As shown in Figure 1, each NDN node maintains three data structures: *(i)* a Content Store (CS) for temporary caching of incoming Data packets depending on local constrains and policies; *(ii)* a routing table named Forwarding Information Base (FIB) used to guide the Interests towards Data; and *(iii)* a Pending Interest Table (PIT), which keeps track of the forwarded Interest(s) that are not yet satisfied with a returned Data packet.

Depending on the application requirements and the access network interface(s), the *Strategy* layer permits to specify different transport and forwarding services, e.g., transmission (and retransmission) of Interests over a single or multiple interfaces simultaneously, management of different Interest priorities.

Forwarding fabric. When a NDN node receives an Interest, the following forwarding fabric is run. First, it searches for a name prefix longest-match in its CS. If a match is found, then the node sends the Data back to the incoming interface of the processed Interest. Otherwise, if there is a matching PIT entry, the Interest is discarded because an equal request has been already forwarded. Otherwise, a new PIT entry is created and the Interest is further forwarded to the interface stored in the FIB, and other actions can be taken according to the devised Strategy.

The Data packet follows the chain of PIT entries back to the requester(s): at the Data reception, every PIT entry is cancelled. If a match is not found in the PIT, then the Data packet is immediately dropped. As a consequence, traffic is *pull-based*: no contents flow unless a consumer has explicitly asked for them and there is a *1-to-1 matching* between Interest and Data. This significantly reduces the amount of unwanted data transfers (e.g., spam) and facilitates the deployment of accountability and control mechanisms on the content routers that manage Interest signaling (e.g., aggregation mechanisms for massive data access, congestion control).

2.2 Related work

NDN is a promising architecture for a general purpose future Internet, including the Internet of Things. Preliminary works in [3] and [5] outline the benefits that a content-centric approach could bring to IoT, including easy data retrieval, the support of mobility, multicast, scalability and security. However, NDN requires some adjustments and customizations to achieve efficient application and traffic performance

in IoT, as argued in [6], where the initial design of a high-level NDN-based architecture for IoT has been proposed.

IoT systems are challenging due to *(i)* multifaceted network environments that may include wired and wireless segments; *(ii)* particular application patterns (e.g., one-to-many, many-to-many) and scope (e.g., locally-relevant data collection in home or on the road); *(iii)* highly heterogeneous device ecosystem where constrained devices coexist with more powerful devices.

Some early works on NDN have investigated peculiar IoT application domains or general optimization strategies for IoT traffic.

Smart building automation has been considered in [7] where the concept of *authenticated Interests* is introduced to allow only authorized applications to control the fixtures.

In [8], the secure configuration of a building management system is presented, while in [9] the initial design of a content-centric homenet is presented and the aspects of naming, node and service discovery are discussed. A secure named data home energy management system is proposed in [10], based on a publish-subscribe layer built on top of NDN to manage groups of fixtures and applications.

In [11] the issue of information freshness in IoT and its impact over NDN caching mechanisms are discussed. The consumer can specify the freshness requirements, and this information is used by the CS to determine if the cached information is suitable or not to satisfy the request.

A traffic optimization strategy to push contents at different granularities is presented in [12]. The authors consider one-way Interests that directly embed small data generated by sensors and are forwarded using the FIB and without creating a PIT entry, because no Data will be sent back. In addition, consumers send a subscription that specifies a sampling period, which is included in the FIB. By doing so, an intermediate node forwards the generated content only if it is compliant with the advertised period.

This paper focuses on a different and still largely unexplored topic: the NDN support of reliable and effective data retrieval from multiple producers in the IoT. This case deserves special attention and opens new issues that need to be managed at the *Strategy* layer of the NDN forwarding fabric, as deeply discussed in the next Sections.

3. THE CASE OF NDN MULTI-SOURCE DATA RETRIEVAL IN IOT

Asking for data from different producers is a common traffic and application pattern in IoT. Consumers may be interested in data items related to a particular event or a geographic area at a certain instant in time without being concerned with *which nodes* produce them. Indeed, in many scenarios:

- the precise *number* and *identity* of producers is not known in advance, e.g., because of the dynamicity and density of the network;
- data are not pre-stored but are *generated on demand* by nodes that host the capability of doing that (e.g., highly volatile data collected in dynamic scenarios);
- multiple different data are requested regardless of their origin to improve the accuracy of the application (e.g., monitoring applications take advantages from data diversity).

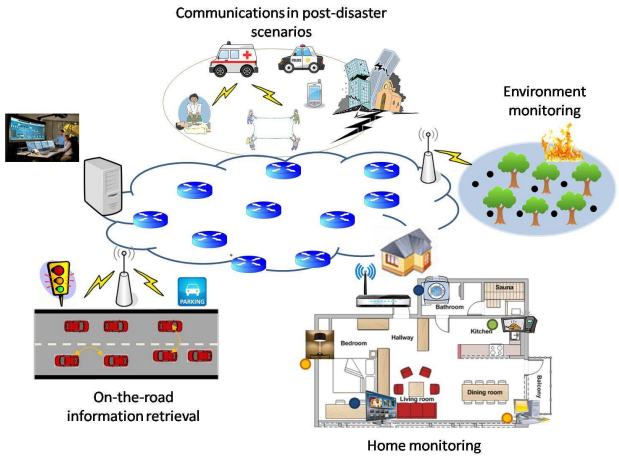


Figure 2: IoT use cases for multi-source data retrieval.

3.1 Use cases

The following use cases, also graphically sketched in Figure 2, clarify how transmitting *a single Interest* to retrieve multiple Data can be beneficially applied either as an efficient alternative to 1-to-1 Interest/Data exchange, when the producers are known, or as the only effective retrieval solution otherwise.

Home monitoring. In a home environment, a unique Home Server (HS) communicates typically wirelessly with a set of end devices (EDs), which may be resource-constrained sensors and actuators embedded in home appliances. EDs act as producers, while the HS is the principal consumer of Data, e.g., it monitors the energy consumption of devices, or the rooms temperature. In alternative, data requests may come from a remote consumer over the Internet, e.g., a utility company monitoring the energy usage of its subscribers. Both consumer and producers are static nodes exchanging small amounts of data (e.g., a measurement accounts for a couple of bytes).

The presence of any ED is known to the HS through a registration procedure where the HS assigns to each ED an identity, an application-specific namespace (e.g., */monitor/temperature* and */monitor/energy* for temperature sensors and devices measuring the energy consumption, respectively), and information for security configuration.

Issuing a single Interest to query a subset of sensors at once, instead of polling them separately, is an efficient solution in order to save network bandwidth.

Environmental monitoring. Wireless sensor networks are typically used for environmental monitoring. To this purpose, a huge number of tiny sensor nodes may be randomly and densely deployed to increase resilience in data collection from a monitored area (e.g., a forest). Sensor nodes take samples (e.g., air quality, relative humidity) over fixed or variable time periods and send them to a local or remote sink (consumer). Both consumer and producers are typically static nodes, unless to consider animals as mobile biological sensors.

The sink may send a single Interest, irrespective of the identifiers and placement of sensors (often unknown, since they are not individually placed), by only specifying the at-

tributes of Data to be retrieved. Received data from different sources can feed the monitoring application for real-time and long-term analytics.

Shortening the data collection time would be particular helpful to promptly infer dangerous conditions, and along with the decrease of the generated traffic load contribute in saving the battery of sensors, which are left unattended for months and even years and may be either unrechargeable or equipped with power scavenging methods, such as solar cells.

Communication in post-disaster scenarios. After a natural disaster crucial information from the incident area has to be collected. Since the infrastructure might be out of action, patrol vehicles may send queries about latest state at various locations towards nodes that crowd-sense the scenario and survey local damage with their mobile devices, e.g., rescue volunteers, private citizens.

Multi-source data retrieval is the only viable solution to collect data from unknown nodes in this highly dynamic environment.

On-the-road information retrieval. Wi-Fi Access Points (APs) city-wide deployed or Road-side Units (RSUs) collect data from passing by vehicles. A remote server may specify the topological scope for relevant data collection. For instance, retrieved data may be road congestion information, pollution measurements from sensors embedded in vehicles, or even photos or videos from the area under surveillance.

An Interest broadcasted by an AP/RSU can request data from vehicles in radio range, without targeting a specific vehicle. Interests may be hop-by-hop relayed by vehicles in a given area of relevance that is wider than the RSU coverage. For instance, an Interest that conveys a name like */traffic/Road101/south/40,41/* requests traffic information about a specified region of Road 101 (southbound, kilometres 40-41). Several vehicles may reply and the larger the number of Data received in a pre-defined (typically short) time interval, the higher the accuracy of the surveillance application (e.g., for inferring traffic jam and promptly give rerouting suggestions to drivers).

Also vehicles may request road traffic information directly to other vehicles instead of relying on a remote server. By doing so, information can be retrieved in a shorter time. As shown in [13] the NDN's request/reply exchange model seems ideally suited to V2V local data exchange.

In summary, issuing a request per multiple Data is the simplest and more natural way to support the aforementioned use cases. The single-Interest solution would decrease the network bandwidth usage and shorten the data retrieval time, by also reducing the energy demands on resource-constrained wireless devices and enabling sleep modes more easily. However, changes at a macroscopic scale in the NDN engine and care in the design of meaningful naming schemes are necessary. Also specific routines should be devised at the *Strategy* layer to tackle the problems uniquely arising over the wireless segment that characterizes all the identified usage scenarios. The main design issues are discussed in the following.

3.2 Open Design Issues

Name diversity. To trigger more Data at once, the Interest should carry a name able to match the content of distinct producers. At the same time, the consumer must distinguish the retrieved Data. Thanks to the fact that the hierarchical URI-like NDN namespace is highly expressive and customizable, a proper naming scheme can be designed to accommodate such a feature.

PIT entries deletion. To support multi-source data retrieval with a single Interest, the PIT management should be rethought. The pending PIT entry should not be deleted upon the first Data arrival, and successive Data packets matching the same Interest should not be discarded.

Data collision. Due to the shared nature of the wireless medium, several producers are expected to reply to the received Interest and likely collide by simultaneously transmitting Data packets. A *distributed collision avoidance scheme* should be devised to reduce the amount of collisions.

Channel unreliability. Due to the inherent unreliability of the wireless channel, Interests and Data are subject to loss and corruption. Hence, *Interest retransmission routines* should be foreseen if the application asks for reliable Data retrieval. The collection of Data from multiple sources implies the definition of new workarounds to manage retransmission timeouts and to recover specific losses.

Data redundancy. A high number of producers could answer the same Interest (e.g., hundreds of sensors with similar collected data could simultaneously reply). A *Data suppression* mechanism is needed to cope with such situations accounting for the demands of the application and the potential constraints of the producers.

4. THE PROPOSAL IN DETAIL

The need for changes in the vanilla NDN implementation in order to support multi-source scenarios such as wireless sensor networks was preliminarily introduced in [14] and [15].

The work in [16] focuses on data collection from vehicles, however the proposal targets the retrieval of a single Data packet per Interest, originated from a vehicle passing nearby the RSU. Therefore, it does not tackle multi-source data retrieval as we defined in this paper.

In [17] a prefix hopping (PH) algorithm is proposed to support multi-source data retrieval in large scale peer-to-peer applications with frequent content updates. PH allows producers to generate multiple content names sharing a generic prefix in a fully distributed manner and it is an alternative solution to the standard NDN EXCLUDE filter, which instead we use in this paper.

In this work, we target *reliable NDN multi-source data retrieval* for IoT systems and focus on *single-hop* wireless scenarios. Such a choice is to dissect the complexity of the design at this first stage of research and then, in a future work, extend the analysis to multi-hop by using a divide and conquer approach. In multihop scenarios data redundancy and network congestion are unavoidably exacerbated. Proper countermeasures should be designed in this case, such as data aggregation at intermediate nodes, clustering schemes that elect multiple collection points inside the network, and so on.

In the following subsections, our proposal is described, by addressing the cases in which data are *(i)* locally processed by the one-hop far consumer, and *(ii)* requested by an application running over a remote host.

The main features of our proposal are summarized in Table 1.

Table 1: Main features of our proposal for reliable multi-source data retrieval

Issue	Solution
Name diversity	A *common name prefix* is included in the name field of a particular Interest, called *msINT*. Replied Data packets include an additional *producer-specific* name component
PIT entries deletion	The PIT entry for a *msINT* is kept until its lifetime expires; multiple Data packets are accepted
Data collision	Defer times are used to space-out Data transmissions from multiple sources
Channel unreliability	The *msINT* is retransmitted as its lifetime expires with an Exclude field indicating producers that already replied
Data redundancy	Overhearing is enforced at the producer-side to cancel Data transmissions

4.1 Local consumer scenario

In the reference *single-hop wireless scenario* a collection point (CP) is interested in the retrieval of N different Data packets from different producers. The number N:

- can be known by the consumer in advance and related to a specific set of producers; e.g., in a home environment the HS knows the number and the type of active data sources;
- can be a value set according to the accuracy demands of the application without a previous knowledge on the number or identity of potential producers, e.g., an RSU looks for data from different cars passing by.

In both cases, to retrieve the N Data items, the CP transmits a particular Interest, in the following referred to as multi-source Interest ($msINT$). The Interest can be simultaneously received by different nodes, some of which can be content producers that answer with a Data packet. To limit the Interest processing to single hop producers, the Interest's SCOPE field is set to 2 while the ANSWERORIGINKIND field specifies that only the original source can answer the request; hence, if a node is not a producer of the requested content, it simply discards the received Interest.

4.1.1 Naming and PIT management

Naming. Since the consumer must distinguish between different Data satisfying the same *msINT*, we assume that contents are associated with a *common name prefix* followed by a *producer-specific* part. Precisely, the first component(s) of the name are the same as those included in the $msINT$, while the last component(s) are related to the producer.

Depending on the application and network scenario, different producer-specific components can be devised, like logical names, random nonces, temporal/geographical information. For instance, in a home monitoring application, the CP sends the Interest */monitor/temperature* to collect data from a set of temperature sensors deployed in the house. The sensor in the kitchen answers with data */monitor/temperature/kitchen*, while the sensor in the bathroom answers with data */monitor/temperature/bathroom*. Vice versa, in case of a monitored environment where devices are not configured with logical names, a random nonce could be used, e.g., */monitor/temperature/0983*.

PIT management. The consumer does not delete the $msINT$ at reception of the first Data packet, but simply caches the received content while maintaining the Interest in the PIT to accept successive Data. Consequently, the Interest lifetime T_{msINT} must be adequately set to accommodate the reception of multiple packets.

When a subsequent Data arrives, the consumer first checks the PIT. If the $msINT$ is not expired, it looks at the Content Store to search for possible duplications. Although the producer-specific name component makes unique the Data of different producers, it is possible that the same producer answers with the same Data (or a fresher version) to subsequent $msINT$. If a CS match is found, cached Data is replaced only if the newly received Data is fresher than the previous one, otherwise the packet is discarded.

4.1.2 Collision Avoidance and Data Suppression

Depending on the expected number of producers, the Data characteristics and the interfering traffic in the scenario, different collision avoidance and suppression schemes can be devised, respectively, to reduce collisions among simultaneously transmitting nodes and to control Data redundancy.

We propose a distributed *consumer-aided* technique that allows the consumer to specify a set of rules for both the collision avoidance and the Data suppression decision to be enforced at the producer-side. Rules are specified in new optional field(s) in the Interest.

Collision avoidance. The collision avoidance scheme simply uses *defer times* to space-out Data transmissions from multiple sources. After receiving the $msINT$, each producer P computes its own collision avoidance timer, τ_{ca}, and waits. At the end of the waiting period, if the Data is still valid (no suppression rule has been executed), P sends the packet.

The simplest collision avoidance scheme is based on a purely *random* computation. The consumer advertises in $msINT$ a *maximum NDN contention window* NCW_{max} and a slot time value V_{slot}.

Each producer extracts a random NCW_r value from a uniform distribution in the range $[0, NCW_{max} - 1]$ and then sets its defer time as $\tau_{ca} = V_{slot} * NCW_r$. Intuitively, the higher the number of potential producers the larger should be NCW_{max}. The consumer may also *dynamically* adjust the NCW_{max} when transmitting subsequent $msINTs$. More details about the *adaptive* NCW_{max} setting will be presented in Section 5.

In addition to a purely random scheme, other (optional) content's attributes could be taken into account for finer tuning of the collision avoidance timer, in order to give higher access priority to those Data that better match the request. For instance, if the consumer specifies a *freshness* parameter, intended as the time a certain packet will be allowed to be held by nodes, then the producers with fresher Data should probabilistically transmit before the others. In such a case, NCW_{max} will be set inversely proportional to the dif-

ference between the freshness advertised by the consumer, as conceived in [11], and the actual Data freshness.

Suppression rules. Optional content's attributes can be used to limit the number of producers answering the $msINT$. For instance, battery-powered producers could decide whether to transmit or not based on their residual charge.

Data suppression can also be decided during the collision avoidance waiting time, if some events occur on the channel while the producer is overhearing. For instance, if the consumer specifies the number N of expected Data packets and the producer overhears N Data transmitted by others nodes, then it can abort its own transmission.

4.1.3 Reliability

In order to collect N Data, the consumer CP waits for a specific time interval, which is equal to the *msINT lifetime*, T_{msINT}. If the lifetime expires and the consumer has not received the expected N Data packets, it retransmits the $msINT$ up to a maximum number of times, n_{msINT}[1]. Specifically, if no content has been received, the consumer retransmits the same $msINT$; otherwise, if M Data packets have been received (with $M < N$), then CP transmits a $msINT$ with an active EXCLUDE field (EF).

In our design, the field contains the list of already received producer-specific components of the content name, e.g., *[kitchen;bathroom]* if we consider the home monitoring example. Therefore, only the producers whose specific name component do not appear in the list will answer the newly issued $msINT$, e.g, in the home example, the originator of the data named */monitor/temperature/bedroom*.

According to the official CCN implementation, *CCNx* [18], a consumer can simply enumerate the elements of the exclusion set, if that is sufficiently compact, or use Bloom filters to reduce the overhead. Therefore, how to actually implement the EXCLUDE field depends on the namespace design and the payload's constraints.

We also observe that the rationale behind the usage of the EXCLUDE field is the same as the one conceived in the legacy NDN architecture, but there is a fundamental difference in the Data retrieval scheme. Traditionally, the EXCLUDE field is set to improve the Interest propagation in the network, by keeping the assumption that *one Interest is consumed by one Data*. As described in [19], more Interests with a proper EXCLUDE field can be transmitted to discover more data of the same type. For instance, in Figure 3, after the consumer receives Data */name/n10*, it will send a new Interest, with the same name prefix but with "n10" in the EXCLUDE field, in order to retrieve more Data items if any exists.

Figures 4 instead summarize the behaviour of our multi-source retrieval algorithm, by considering the case when no retransmission occurs (a) and when the consumer retransmits the Interest with EXCLUDE field (b).

4.1.4 msINT lifetime setting

The $msINT$ lifetime depends on the time required to collect N Data and must be set to accommodate transmissions by the producers even in the worst-case, i.e., when considering the highest values of the collision avoidance timer, $\tau_{ca,max}$, and of the channel access delay, $maxAccessWaitTime$.

[1]Parameter n_{msINT} can be set to reflect the time-to-live (TTL) of the Data, so that $(n_{msINT} \cdot T_{msINT}) \leq$ TTL.

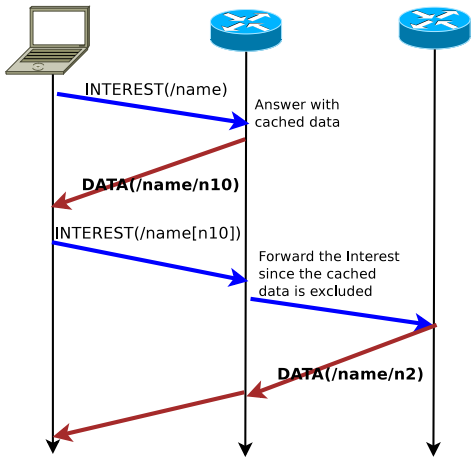


Figure 3: Traditional NDN data retrieval with the EXCLUDE field.

T_{msINT} is set as follows: $\tau_{ca,max} + maxAccessWaitTime$. The $maxAccessWaitTime$ parameter depends on the specific MAC layer technology and can be somehow inferred by the consumer[2].

4.2 Remote consumer scenario

The local scenario can be generalised to the case of a remote consumer (RC) that asks for Data to the collection point (CP), which then interacts with the wireless producers. The RC sends the so-called Long-Lived-Interest (LLI). Initially conceived in [19] and [20], the LLI is an Interest with a long lifetime in order to retrieve more Data with the same name in different time instants. The LLI acts as a subscription that avoids continuous Interest refresh and reduces the signalling overhead.

After receiving the LLI, the CP can decide to immediately answer with cached Data packets if any or it can build a $msINT$ that is broadcasted to collect the Data as described in the previous section. The CP may autonomously decide how many times the $msINT$ is retransmitted to retrieve all Data, without involving the RC.

Depending on the size, Data packets retrieved in the wireless segment can be *aggregated* in a single packet that does not exceed the path Maximum Transfer Unit (MTU) and sent back to the RC, or they can be individually transmitted. Moreover, if the application does not need individual reports, the CP can also perform filtering operations and/or compute local statistics and then send only the resulting Data back to the CR.

We use distinct names for $msINT$ and LLI for two main reasons:

1. the LLI usually refers to the retrieval of *multiple Data* from *the same content source* at different time instants, while $msINT$ is intended for retrieval from *different sources* more or less simultaneously;

[2]For the 802.11 access technology, $maxAccessWaitTime$ can be set to $W * (\sigma + T_{maxPkt})$, where W is the MAC-layer contention window size for broadcast transmissions, σ is the time slot duration, and T_{maxPkt} the time to transmit a packet of the largest size at the basic rate.

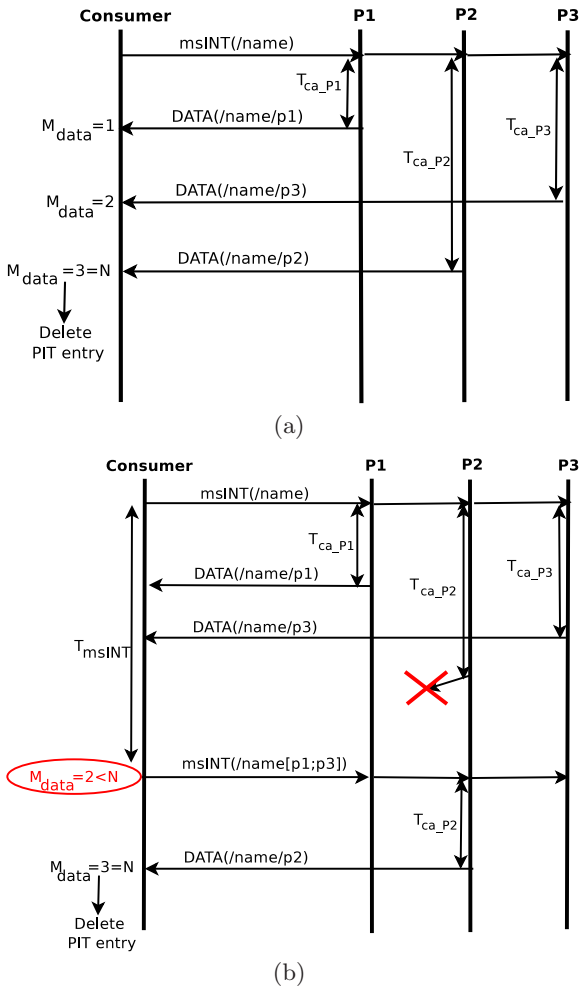


Figure 4: Flow chart of exchanged packets without retransmissions (a) and with retransmission and EXCLUDE **Field (b).**

2. the *LLI* lifetime is longer than the *msINT* lifetime: it must accommodate the overall Data collection time plus the round-trip-delay between the RC and CP.

5. PERFORMANCE EVALUATION

To assess the performance of the proposed multi-source data retrieval framework, we used the Network Simulator 3 (ns-3) and the open-source ndnSIM [21] module, specifically designed to support NDN. We properly overhauled ndnSIM to implement the conceived routines.

We consider two distinct cases for performance evaluation: *(i)* a Home scenario, where the CP is also the local consumer and it queries producers under its control; *(ii)* a Vehicular scenario where a remote consumer asks the CP to gather information from nearby sources, whose identity and number is not a priori known.

The main parameters settings are summarized in Table 2. Simulation results, averaged over 10 independent runs, are reported with the 95% confidence intervals.

Table 2: Simulation settings

Scenario	Parameter	Setting(s)
Home Monitoring	Collection pattern	periodical
	Total monitoring time	3 hours
	Monitoring interval	5 minutes
	Data Payload	128 byte
	Producers (N)	4-16
	Interfering traffic	CBR (20 Interest/s)
	Access technology	IEEE 802.11g
	Propagation model	Rayleigh
	Coverage radius	$\approx$ 150 m
Vehicular Traffic Control	Collection pattern	occasional
	msINT retransmissions	0-3
	Data payload	1024 byte
	Vehicles (N_v)	40;80
	Interfering traffic	None
	Access technology	IEEE 802.11p
	Propagation model	Nakagami
	Coverage radius	$\approx$ 300 m

5.1 Home scenario

The first set of results has been collected when considering a home network with an area size of $100m^2$. The CP is the home server that periodically (every 5 minutes) queries sensors to retrieve temperature measurements so to regulate air conditioning/heating in each room in a differentiated way.

We assume that a variable number of devices, N, ranging from 4 to 16, acts as producers by generating 128 bytes-long Data. Since the CP knows the identity and number of devices under its control, it will retransmit the $msINT$ until the collection process is complete at every period.

For realistic evaluation, we consider the Rayleigh signal propagation and we model the interfering traffic as a constant bit rate (CBR) application with 20 Interests per second and Data packets of 1024 bytes. Therefore, packets could be lost due to collisions and adverse propagation effects.

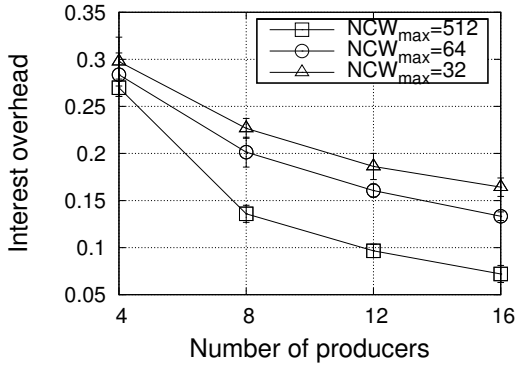
As key performance indicators (KPI), we consider: *(i)* the *Interest overhead*, defined as the average ratio between the number of Interest sent by the CP and the received Data packets, at every collection round; *(ii)* the *collection time*, defined as the average time to complete the collection of N Data packets at every collection round; and *(iii)* the *Interest/Data number*, which is the overall number of Interests/Data transmitted/received by the CP during the entire monitoring time.

In Figure 5, we evaluate the performance of the collision avoidance routine of our multi-source data retrieval framework, in the following referred to as single Interest-multiple Data (SIMD), by varying the NCW_{max} parameter. The SIMD slot time V_{slot} is set equal to the IEEE 802.11g DIFS parameter.

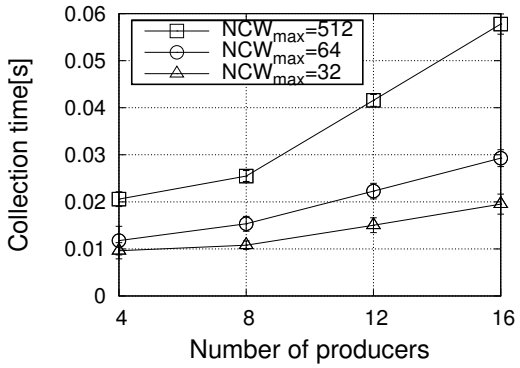
Results are highly sensitive to the NCW_{max} setting.

The Interest overhead gets significantly lower (about halved) when NCW_{max} passes from 32 to 512 and the number of producers increases, since larger NCW_{max} are more effective to counteract more likely collisions. Interestingly, as the number of producers increases, the Interest overhead decreases; this is an intrinsic property of the SIMD scheme.

However, the bandwidth saved with a larger NCW_{max} is paid in terms of a longer collection time, that increases as the number of producers increases since more retransmissions should be performed to recover losses from a higher number of nodes. Recovery operations are slowed, since T_{msINT} gets longer as NCW_{max} increases.



(a) Interest overhead



(b) Collection Time

Figure 5: Home scenario: metrics vs. the number of producers when varying the size of the NDN contention window for the SIMD scheme.

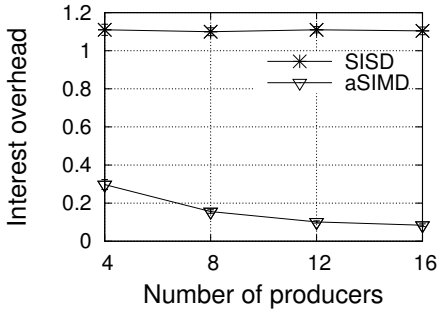
In the considered scenario, since the CP knows how many nodes (N) are expected to reply to the issued $msINT$, a smarter solution would be to *adaptively* set the NCW_{max} parameter and advertise it in the Interest, so not to waste time and to limit collisions. Intuitively, a larger NCW_{max}, i.e., 512, is used when the number of nodes is higher; on the contrary, shorter values, i.e., 32, 64, are preferred in subsequent Interest retransmissions as the residual number of producers meant to reply is reduced.

In Figure 6, the performance of the adaptive SIMD is compared against the native NDN Single Interest-Single Data (SISD) retrieval scheme. In SISD, the Interest overhead is almost constant as the number of producers increases, nearly equal to 1.1, and heavily higher than the value experienced by adaptive SIMD (around 0.05-0.3), Figure 6(a). In particular, for 16 consumers, the overhead decreases by one order of magnitude.

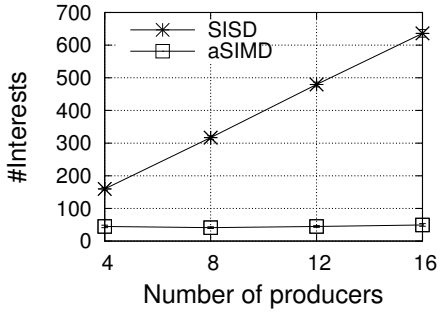
The claimed efficiency of SIMD compared to SISD can be also observed in Figure 6(b): the number of Interest is extremely low and almost unsensitive to the number of producers. In addition, regardless of the number of producers, adaptive SIMD outperforms SISD also in terms of collection time, reduced of nearly 10ms.

5.2 Vehicular scenario

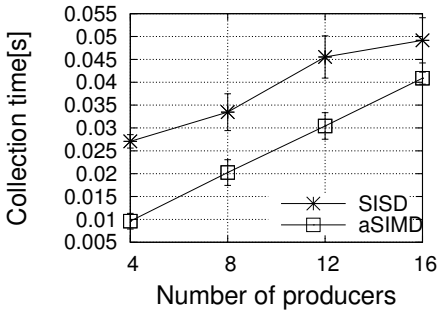
The second set of results considers a traffic congestion control application. The collection point is an RSU lo-



(a) Interest overhead



(b) Number of Interest packets

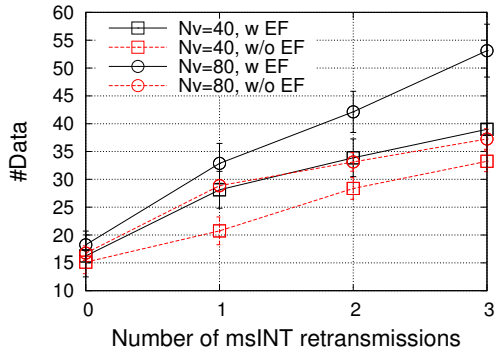


(c) Collection Time

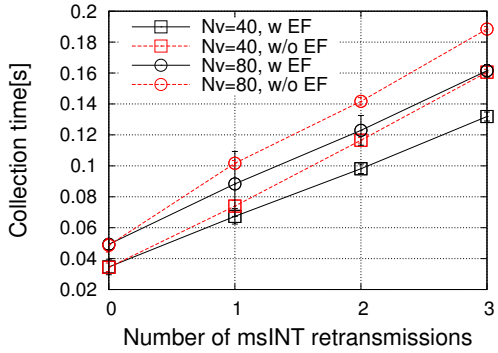
Figure 6: Home scenario: metrics vs. the number of producers when comparing the SISD scheme against the adaptive SIMD (aSIMD) scheme.

cated in the central reservation of the highway "A1", consisting of a dual carriageway with three traffic lanes per each direction. A remote control center (RCC) is interested in the retrieval of the average vehicles speed in the north carriageway at kilometres 20-21 in order to estimate the overall conditions of the traffic. Each vehicle is equipped with a global positioning system (GPS) device and an IEEE 802.11p transceiver, and is able to store its kinematics (speed, acceleration) and position parameters (location, direction, and route number along with a timestamp).

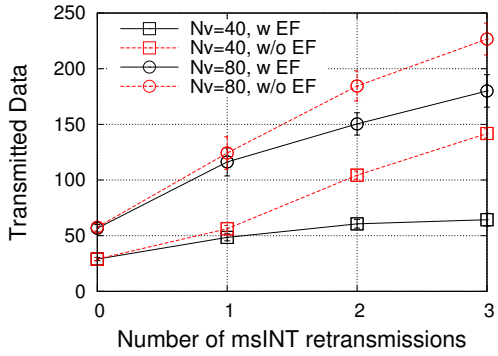
The retrieval process is initiated by the RCC, which sends an LLI with name */traffic/speed/highwayA1/North/{20,21}*. At the reception of the LLI, the RSU broadcasts a $msINT$ to collect the kinematics parameters of as much as possible vehicles passing by. On receiving the $msINT$, vehicles in the North carriageway will schedule Data transmission by following the designed collision avoidance strategy. Once a consistent number of speed samples has been collected, the



(a) Number of collected (distinct) Data packets



(b) Collection Time



(c) Total number of transmitted Data packets

Figure 7: Vehicular scenario: metrics vs. the number of msINT retransmissions when varying the number of producers Nv for the SIMD scheme with and without the EXCLUDE field, NCW_{max}= 127, V_{slot} =IEEE 802.11p DIFS.

RSU calculates the average vehicle speed and sends a Data packet to the RCC. The RSU is oblivious to the number and the identity of vehicles in its transmission range. Therefore, it can retransmit the $msINT$ for a predefined number of times to maximize the number of collected Data.

As KPI, we consider the *number of Data collected by the RSU* and the correspondent *collection time*.

In the simulation, we vary the number of $msINT$ retransmissions (from zero - no retransmission occurs - to three), and we consider two distinct number of producer vehicles, 40 and 80.

In Figure 7 a simplified SIMD scheme (curves are labeled as *w/o EF*) is used as a benchmarking solution. An Interest is issued to retrieve Data at once and it does not foresee the use of the EXCLUDE field in retransmitted Interest. As a consequence, at every retransmission the same producers may attempt to reply and likely collide. For both schemes, as expected, the higher the number of $msINT$ retransmissions the higher the number of collected Data. Results clearly show that the retransmissions of the $msINT$ with the EXCLUDE field carrying the number of already received Data packets lead to the following benefits. The consumer can increase the number of different Data packets collected in a shorter time because a lower number of nodes is authorized to reply to subsequent retransmitted $msINT$. As a matter of fact, in presence of 40 producers, SIMD with the EXCLUDE field lets the RSU retrieve the same number of Data that is retrieved with 80 producers when the scheme without EF is considered, hence confirming the adequacy of the proposed solution to maximize data diversity.

The overall number of Data packets transmitted by producers is also reduced (Figure 7(c)), with a consequent benefit in terms of network bandwidth and device resources saving. It is worth noticing that the latter aspect may be especially advantageous for constrained devices, which can go back to a sleeping state once receiving the $msINT$ carrying their producer component in the EXCLUDE field, and hence acting as an implicit acknowledgement.

As a further notice, the Interest size may increase when the EXCLUDE field is conveyed. However, Bloom filters can be used to keep the incurred overhead under control[3].

6. CONCLUSION

In this paper we have investigated a common IoT traffic pattern, multi-source data retrieval, supported through named data networking. After scanning related IoT scenarios and identifying their peculiarities, the study pinpointed the potential benefits of NDN and open design issues, mainly involving the naming scheme design and some re-engineering in the forwarding fabric and retransmission routines.

The work designs a comprehensive framework for *reliable data retrieval* from *multiple wireless sources* from *local and remote consumers*. It tackles identified problems, through *(i)* a distributed consumer-aided collision avoidance scheme, *(ii)* a Data suppression mechanism to reduce packet redundancy, *(iii)* and the EXCLUDE field added in Interest packets to manage selective retransmissions. Simulation results in ndnSIM confirm the benefits of the conceived solution, that save bandwidth, while maximizing data diversity and shortening the collection time.

The conducted study would also provide some hints for future research. Indeed, the conceived solution is a *baseline framework* on top of which further modules may be designed for collision avoidance, suppression and retransmission to be customized according to the requirements of different applications, networks and devices. As a future work, we plan to extend the study to multi-hop wireless scenarios, where issues may be exacerbated and additional workarounds need to be devised.

[3]In [22] it is shown that for less than 100 components, with a filter length of 128 bytes, the false positive probability is below 1%.

7. ACKNOWLEDGEMENT

This work has been carried out within the national research project PON03PE_00050 DOMUS "Home automation systems for a cooperative energy brokerage service".

8. REFERENCES

[1] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: a survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[2] L. Zhang *et al.*, "Named Data Networking (NDN) Project," PARC, Tech. Rep. NDN-0001, October 2010.

[3] Z. Sheng, S. Yang, Y. Yu, A. V. Vasilakos, J. A. McCann, and K. K. Leung, "A Survey on the IETF Protocol Suite for the Internet of Things: Standards, Challenges, and Opportunities," *IEEE Wireless Communications Magazine*, vol. 20, no. 6, pp. 91–98, 2013.

[4] V. Jacobson *et al.*, "Networking Named Content," in *ACM CoNEXT*, 2009.

[5] Y. Zhang, D. Raychadhuri, R. Ravindran, and G.-Q. Wang, "ICN based Architecture for IoT," in *Internet-Draft*, June 2014.

[6] M. Amadeo, C. Campolo, A. Iera, and A. Molinaro, "Named Data Networking for IoT: an Architectural Perspective," in *European Conference on Networks and Communications (EuCNC)*, Bologna, Italy, 2014.

[7] J. Burke, P. Gasti, N. Nathan, and G. Tsudik, "Securing Instrumented Environments over Content-Centric Networking: the Case of Lighting Control and NDN," in *IEEE INFOCOM NOMEN Workshop*, 2013.

[8] W. Shang, Q. Ding, A. Mariantoni, J. Burke, and L. Zhang, "Securing Building Management Systems Using Named Data Networking," *IEEE Network*, vol. 28, no. 3, pp. 50–56, 2014.

[9] R. Ravindran, T. Biswas, X. Zhang, A. Chakraborti, and G. Wang, "Information-Centric Networking Based Homenet," in *IFIP/IEEE ManFI Workshop*, 2013.

[10] J. Zhang, Q. Li, and E. M. Schooler, "iHEMS: an information-centric approach to secure home energy management," in *IEEE SmartGridComm*, 2012.

[11] J. Quevedo, D. Corujo, and R. Aguiar, "Consumer Driven Information Freshness Approach for Content Centric Networking," in *IEEE INFOCOM NOM Workshop*, Toronto, Canada, 2014.

[12] J. François, T. Cholez, and T. Engel, "CCN Traffic Optimization for IoT," in *The 4th International Conf. on Network of the Future (NoF)*, 2013.

[13] L. Wang, A. Afanasyev, R. Kunts, R. Vuyyuru, R. Wakikawa, and L. Zhang, "Rapid Traffic Information Dissemination Using Named Data," in *ACM NoM Workshop*, 2012.

[14] N.-T. Dinh and Y. Kim, "Potential of Information-Centric Wireless Sensor and Actor Networking," in *IEEE International Conference on Computing, Management and Telecommunications (ComManTel)*, 2013.

[15] J. P. Meijers, M. Amadeo, C. Campolo, A. Molinaro, S. Paratore, G. Ruggeri, and M. Booysen, "A Two-Tier Content-Centric Architecture for Wireless Sensor Networks," in *IEEE ICNP*, Gottingen, Germany, 2013.

[16] J. Wang, R. Wakikawa, and L. Zhang, "DMND: collecting data from mobiles using named data," in *IEEE Vehicular Networking Conference (VNC)*, 2010, pp. 49–56.

[17] F. Angius, C. Westphal, J. Wei, M. Gerla, and G. Pau, "Prefix Hopping: Efficient Many-To-Many Communication Support in Information Centric Networks," in *IEEE INFOCOM NOMEN Workshop*, 2013.

[18] "CCNx Project, http://www.ccnx.org."

[19] Z. Zhu, S. Wang, X. Yang, V. Jacobson, and L. Zhang, "ACT: audio conference tool over named data networking," in *ACM SIGCOMM workshop on Information-centric networking (ICN)*, 2011.

[20] A. Carzaniga, M. Papalini, and A. L. Wolf, "Content-Based Publish/Subscribe Networking and Information-Centric Networking," in *ACM SIGCOMM workshop on Information-centric networking (ICN)*, 2011.

[21] A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnSIM: NDN simulator for NS-3," NDN Project, Tech. Rep. NDN-0005, July 2012.

[22] S. Tarkoma, C. E. Rothenberg, and E. Lagerspetz, "Theory and practice of Bloom filters for distributed systems," *IEEE Communications Surveys and Tutorials*, vol. 14, no. 1, pp. 131–155, 2012.