Design of Information-centric publish-subscribe mechanisms for Internet of Things

Department of Electrical and Information Engineering
ELECTRICAL AND INFORMATION ENGINEERING
Ph.D. Program
SSD: ING-INF/03–TELECOMUNICAZIONI

Final Dissertation

Design of Information-centric publish-subscribe mechanisms for Internet of Things

---

by
VENTRELLA Agnese Vincenza

Supervisor:

Prof. GRIECO Luigi Alfredo

Coordinator of Ph.D. Program:
Prof. GRIECO Luigi Alfredo

Course n°31, 01/11/2015-31/10/2018

Department of Electrical and Information Engineering
ELECTRICAL AND INFORMATION ENGINEERING
Ph.D. Program
SSD: ING-INF/03–TELECOMUNICAZIONI

Final Dissertation

Design of Information-centric publish-subscribe mechanisms for Internet of Things

---

by

VENTRELLA Agnese Vincenza

Agnese Vincenza Ventrella

Referees:

Prof. MOLINARO Antonella

Prof. XU Changqiao

Supervisor:

Prof. GRIECO Luigi Alfredo

Coordinator of Ph.D Program:

Prof. GRIECO Luigi Alfredo

Course n°31, 01/11/2015-31/10/2018

Abstract

Emerging Internet of Things (IoT) applications, such as Intelligent Transport System (ITS), require efficient content dissemination mechanisms based on the publish-subscribe model in static and mobile scenarios. The Information-Centric Networking (ICN) architecture can successfully satisfy these requirements. In its native formulation, ICN can fulfill publish-subscribe data dissemination and natively support mobile applications. At the time of this writing, several ICN-based solutions have been proposed to implement the publish-subscribe model, but none of them is explicitly tailored to mobile scenarios. To bridge this gap, this thesis presents the following contributions: (i) new pull-based and push-based publish-subscribe communication schema, able to support user mobility in ICN networks; (ii) analytical models describing the communication overhead they incur, (iii) evaluation of the accuracy of the proposed models through computer simulations. The conducted study considers well-known benchmark network topologies, real IoT monitoring services, and standardized settings for urban and rural environments. From one side, our obtained results validate the conceived analytical models. From another side, they highlight pros and cons of pull-based and push-based approaches by emphasizing the conditions under which one scheme should be preferred to the other one.

Acknowledgements

First and foremost I want to thank my advisor Prof. Luigi Alfredo Grieco for making this possible through his guidance, motivation, and immense knowledge.

My sincere thanks to Dr. Giuseppe Piro for his patience, efforts, and dedication during these years.

Some special words of gratitude go to Dr. Flavio Esposito for his guidance, hard work, and passion for research.

I am also grateful to the reviewers, Dr. Antonella Molinaro and Dr. Changqiao Xu, for their useful suggestions which helped me improve this thesis.

I also wish to thank my fellow labmates for sharing joy and sorrow of my PhD journey.

Last but not the least, I would like to thank my family and my friends.

Contents

Introduction

The current Internet architecture was born in 1971 as an academic network, allowing the communication among the scientific community. In a few years, its usage radically changed: it is now a global infrastructure for a massive distribution of information. Cisco [1] reports that the global Internet Protocol (IP) traffic had an annual run rate of 1.2 Zetabytes in 2016 and it will reach an annual rate of 3.3 Zettabytes in 2019.

The current Internet is based on a host-centric communication mechanism. This means that a client can fetch a given content only if it knows the Uniform Resource Locator (URL) or the IP address of the server that can provide the desired information. But the new users' behavior asks for something different: users are interested in receiving contents regardless of their location in the network, rather than in accessing a particular end host. In other words, "what" users search is more important than "where" these contents are located, because Internet is now an information container in which users put and retrieve contents [2]. This object-location binding can cause problems when the object is relocated within a site, it moves to a different site, the site changes domain or it is temporally/permanently unreachable. Furthermore, copies of the same object can appear at different locations of the system, i.e. when they are available from different web servers accessible through different Uniform Resource Identifier (URI) [3].

As a consequence, Internet is called to better support the dissemination and the retrieval of information, not only connection to a host. The main requirements to guarantee are [4]:

- Quickly, reliable and efficient contents availability;
- Identification of content with location-independent names;
- Security service without any reference to location and to the provider of the content;
- Mobility support, in order to avoid service interruption during access networks changing;
- Scalable management of storage, bandwidth, and computation of service providers in case of huge number of users;
- High fault-tolerance of information and communication technology.

To cope with the constant Internet evolution and to reduce the complexity introduced by cumbersome patches and middleware layers, the scientific community invested millions on new Internet architectures redesign. The

ICN paradigm [5][6] emerged as one of the most promising approach to accomplish this revolution. ICN has been defined as the third revolution in the telecommunication networks [5]: the first is the public switched telephone network that connected wires, the second was the all-IP network that connected nodes and the third is ICN that connects information.

Researchers have designed several ICN architectures, such as Data-Oriented Network Architecture (DONA) [7], Network of Information (NetInf) [8], Publish Subscribe Internet Technology (PURSUIT) [9], MobilityFirst [10], COntent Mediator architecture for content-aware nETworks (COMET) [11] and Named-Data Networking (NDN) [12]. They share some key characteristics. On one hand, a producer announces/registers the generated contents to the system. On the other hand, a consumer sends a request for some contents it is interested in. When the network matches requests and contents, the contents are sent back to the consumer, according to a routing algorithm.

One step ahead in the research about ICN, and the Future Internet in general, is the application and the customization of these new communication primitives to complex and real scenarios. In particular, the IoT technology that allows an ubiquitous connectivity among smart devices deployed worldwide, led to the emergence of novel applications in smart health, smart transportation, smart metering, smart city, and smart buildings [13]-[16]. Most of IoT applications share three main characteristics: first, services are inherently information-centric [17]-[23]. Consumers are more interested in retrieving contents generated by IoT devices rather than communicating with a specific node [17]. Second, IoT contents may be, at the same time, real-time and event-triggered. Therefore, to avoid the continuous polling of a given data producer, contents' updates should be disseminated by using the publish-subscribe communication paradigm. Accordingly, producers publish information objects under a specific topic; consumers express their interest by issuing subscriptions; when a new information object is generated, it is sent to the list of subscribers [24]-[26]. Between producers and consumers, a scalable data dissemination architecture is required to match published topics with subscriptions and send notifications. In addition, consumers and producers may be mobile [27]-[29].

Thesis contribution. Based on these premises, this thesis intends to address the design and the evaluation of ICN-based publish-subscribe communication schemes for IoT applications and specifically ITSs. To this end, two different approaches are formulated: *pull-based* and *push-based* (they extend the baseline solutions already proposed in the literature, i.e., [30]-[34], but for static environments). The former assumes that the user continuously polls the remote server for retrieving any update of the desired content. The latter one, instead, intends to establish a stable communication channel with the remote server, that delivers contents without being continuously solicited. In both cases, the maintenance of a communication path between producer and consumer leads to a not negligible communication overhead, due to control messages exchange and extra data transmission.

In summary, this thesis offers the following contributions:

- baseline pull-based and push-based approaches available in literature are extended to support the mobility of consumers and producers;
- the communication overhead incurred by mobility support is evaluated through analytical models;
- the analytical models are validated through computer simulations in realistic scenarios;
- pros and cons of pull-based and push-based approaches are discussed in all the investigated scenarios.

In particular, the average communication overhead is investigated in urban and rural scenarios, where node density and user speed are set according to [41], the ICN network topology is generated by Boston university Representative Internet Topology gEnerator (BRITE) [35], and many real IoT services (including home security system, health sensors, smart meter, traffic sensors) are taken into account. Simulation results, obtained trough our own ad-hoc simulator written in C++, clearly demonstrate that the average communication overhead increases with the number of nodes belonging to the network topology (specifically, urban scenarios reach the higher communication overheads). Moreover, in a scenario with mobile consumer and stationary producer, both pull-based and push-based approaches experience higher communication overheads when the user speed increases. Same considerations can be done in scenarios where both consumer and producer are mobile. Differently from the previous case, however, a lower communication overhead is registered. On the contrary, in a scenario with stationary consumer and mobile producer, such kind of behavior is only registered for the pull-based approach. With respect to the investigated IoT services, the analysis shows that the communication overhead always increases with the content generation rate, except for the case where the push-based approach is adopted in a scenario with stationary consumer and mobile producer (in which all the IoT applications reach the same communication overhead). Finally, the study highlights that the pull-based approach always guarantees the lowest communication overhead. To provide a further insight, the accuracy of the conceived analytical models is evaluated by calculating the absolute relative error. Resulting values, always less than 10%, demonstrate the good fit of the conceived models.

Thesis structure. Chapter 1 focuses on the challenges in the ITSs, the problems of the current Internet architecture, and how the ICN paradigm can efficiently face them. Chapter 2 presents an overview of the state of the art, with particular reference to the ICN and publish-subscribe communication schema. Section 3 provides the analytical models that catch the overhead incurred by the publish-subscribe scheme in mobile ICN deployments. Simulations and theoretical results are presented and compared in Chapter 4. Finally, the Conclusion section provides closing remarks and draws future research activities.

Chapter 1

Background on IoT & ICN

In this chapter, IoT challenges and Transmission Control Protocol (TCP)/IP architecture problems are investigated. Then, the emergent ICN paradigm is discussed as a solution to the aforementioned problems. Finally, a brief overview of the ICN advancements in the real world is presented.

1.1 Challenges in the Internet of Things

The IoT is a key paradigm for enabling the vast amount of networked objects that offer services, such as sensing, monitoring and actuating [36]. Latest reports from key stakeholders of the telecommunication market forecast that up to 100 billion of connected things are expected to join the IoT by 2020 [1] [37] and several novel applications are emerging in smart health, smart transportation, smart metering, smart city, and smart building [13]-[16]. The IoT definition provided by the International Telecommunication Union (ITU) is the following: "a global infrastructure for the Information Society, enabling advanced services by interconnecting (physical and virtual) things based on, existing and evolving, interoperable information and communication technologies"[38]. IoT systems are characterized by the following requirements [17][36]:

- autonomy: IoT devices are usually low-power embedded systems; therefore, the communication overhead needs to be minimized to extend their system lifetime [36];
- adaptability: the continuous innovation and evolution of the IoT devices requires the system to easily and quickly adapt to the change [39];
- scalability: the high number of devices and produced data that reaches the order of billions [1] needs efficient management;
- robustness: IoT platforms can experience service interruption because of the dynamicity of the environment [40]; therefore, the system has to be resilient to failures;
- traffic characteristics: real-time, event-triggered, asynchronous and periodic traffic has to be efficiently supported [34];

- mobility: sensing and actuation capabilities can also be integrated in mobile devices, such as smartphones, tablets, cars [41]; all the correlated problems due to mobility must to be handled;

- interoperability: the heterogeneity and complexity of IoT devices require the decoupling between high-level applications and physical resources.

Several IP-based standards were proposed by working groups, such as IPv6 over Low power Wireless Personal Area Networks (6LoWPAN), Constrained RESTful Environments (CORE), Routing over Lossy and Low-power Networks (ROLL) [42] to face the IoT demand. Unfortunately, such a classic host-centric IP model links service provisioning to node locations and, consequently, incurs well known issues in handling seamless mobility and multicast data dissemination [43].

Use case: Intelligent Transportation System

In particular, an interesting application of the IoT technology is the ITS. The massive spread of transportation systems closed the gap, but also led to new problems, such as frequent congestion on highways and urban centres, CO_2 emissions dangerous for the environment, waste of energy, and high rates of accidents. Researchers are continuously studying how to cope with these issues making the classic transportation systems evolving towards an **Intelligent** Transportation System and, thus, revolutionizing the way we travel today [14]-[45].

ITS [46] is the application of Information and Communication Technologies (ICT) to the transport sector [47]. The ITS concept was born in the United States (US) in the 20th century, and then spread worldwide. In general, ITS may refer to all mode of transport, but the European Union (EU) defined its application in the field of road transport. In this context, sensing, analysis, control, and communications technologies are applied to ground transportation in order to improve efficiency, sustainability, safety, mobility, environment impact, and comfort. The first generation of ITS present in the marketplace includes Advanced Driver Assistance Systems (ADAS) for cars, electronic tolling systems, and traffic information systems [48]. The limitation of these systems is that they are stand-alone, because they cannot share data and cooperate. The next generation of ITS is the Cooperative Intelligent Transport Systems (C-ITS). Here, the involved entities communicate and share information. This field is currently under research [49]. The typical architecture used in the ITS context is Vehicular ad hoc networks (VANET) [50]. VANET [51] includes intelligent vehicles equipped with transceivers and on board applications, Road Side Unit (RSU), centralized management system, communication links etc,.

This kind of system has the following features [19]:

- Information-oriented data: ITS data refer to a content (e.g. traffic jam) and are generated in a massive amount. The identity of the data producer is not relevant and a group of user can be interested in the retrieval of such information.

- Mobility: the system is composed of fixed nodes and mobile nodes that can move at variable speed. This leads to a constantly changing network topology. Moreover, ITS requires wireless communications among vehicles and between vehicles and the road side infrastructure. In this scenario, the inherent unstable connectivity of the wireless channel is exacerbate by the high mobility.

- Security: exchanged data need to be trusted, because of the dangerous consequence of possible attacks.

The ITS architecture has been standardized by various organisations (e.g International Standards Organization (ISO) [52], European Telecommunications Standards Institute (ETSI) [53]), but they are all based on the classic host-centric IP network layer. A relevant number of works [19]-[55] state that the current Internet architecture is not able to effectively fulfil all the aforementioned requirements.

1.2 Shortcomings of the TCP/IP Architecture

The current use of the TCP/IP protocol is far away from the original goal it was designed. The initial network was composed of fixed and trustworthy hosts. It lacked of any kind of security mechanisms. The expansion of the network, the introduction of mobile nodes, the variety and amount of generated contents (voice, audio broadcasting, and video) raised the necessity for patches, add-ons, middleboxes, overlays, and so on [5][56].

One of the biggest problem of the TCP/IP architecture has been identified in the addressing [56]-[58]. IP addresses are assigned to the interfaces and not to the applications. Since 1972, this problem was recognised: when researchers tried to add more connections to the same host for redundancy, it did not work because the traffic that cannot reach an interface does not know that it can reach the same host through a different interface, but it looks like a different host.

Addressing the node can sound like a good solution, but it is not enough because what we look for is the application on the node. Moreover, the application has to be reachable even if it moves to a different host. In 1978, John Shoch defined the concept of name, address and route [58]:

- name: the name of a resource refers to "what we seek";

- address: the address refers to "where it is";

- route: the route "tells us how to get it".

Even if the problem was clear, the Internet still does not use names. We mention herein some of the issues of the current Internet and how they have been handled, with the addition of a significant amount of complexity:

- Information delivery: a user is interested in a content and not in the location where it can retrieve the desired data. Therefore, a content should be available in the shortest time from the best available source and not necessarily from the host that generated it. This is feasible today thanks to the Content Delivery Networks (CDNs) [59], but this solution requires to add a significant amount of complexity to the basic model. Proprietary and costly overlay solutions typically employ network-unaware mechanisms which lead to inefficient utilization of the underlying network resources;

- Mobility: Cisco [1] estimated that there will be 11.6 billion mobile-connected devices by 2021, including Machine-to-Machine (M2M) modules. This number exceeds the world's projected population of 7.8 billion. Mobile IP, and then Mobile IPv6, are examples of protocols a posteriori introduced to allow mobile devices to move from one network to another while maintaining a permanent IP address [60];

- Security: the Internet was designed to operate in a completely trustworthy environment. But this is not the case anymore. Therefore, add-on security patches, trust mechanisms, new security protocols that complement the existing (inter)networking protocols (e.g., IPSec and DNSSec) have been introduced.

1.3 Information Centric Networking

In front of the changes in the Internet usage and of the resulting raised problems, part of the scientific community started designing new architectures [6]. In this context, the ICN paradigm emerged as on the most promising approaches. Differently from the TCP/IP model, ICN is based on the assumption that (i) users, programs, and hosts are untrustworthy and mobile; (ii) communication is often multiaccess; (iii) users are interested in retrieving, processing, and sharing information [5].

Several architectures have been designed, such as DONA [7], NetInf [8], PURSUIT [9], Named-Data Networking (NDN) [12], and COMET [11]. All of them share key characteristics, but implemented with different approaches. The common operating principle is shown in Figure 1.1. On one hand, a producer announces/registers the generated contents to the system. On the other hand, a consumer sends a request for some contents it is interested in. When requests and contents are matched by the network, the contents are sent back to the consumer, according to a routing algorithm. The main ICN architectures are presented in what follows.

Named-Data Networking (NDN)
NDN leverages a receiver-driven architecture and only two kind of messages are allowed: *Interest* and *Data* packets. The communication process envisages that: (1) the user, that wants to retrieve a content, issues an *Interest* packet; (2) the network delivers the user request towards the node able to provide the corresponding answer; (3) the content is stored within a *Data* packet and is sent back to the user through the reverse path of the *Interest* packet. Now,

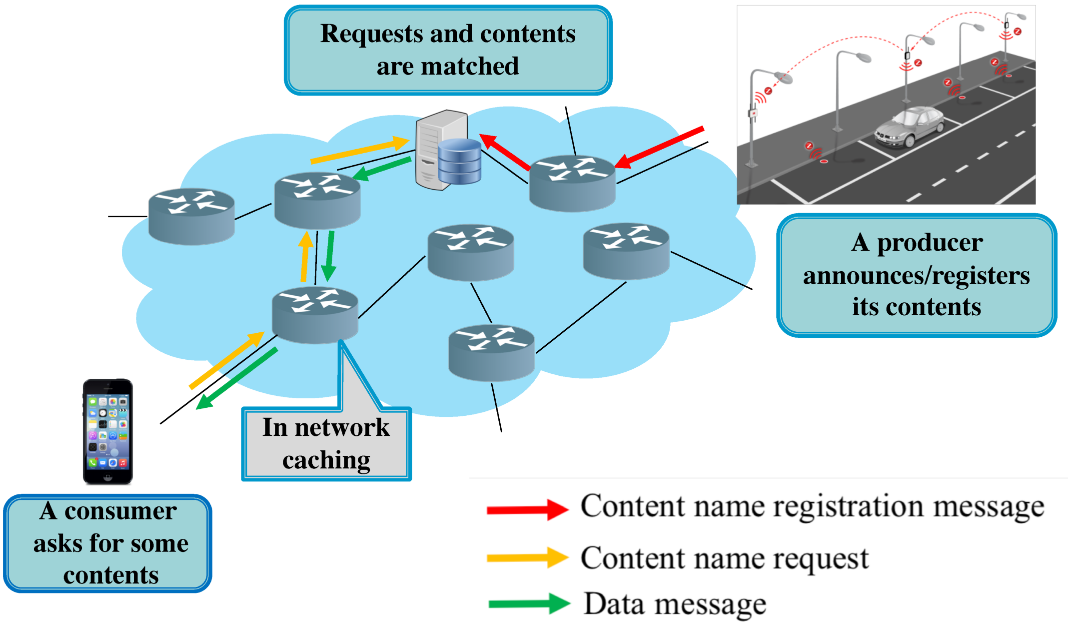


FIGURE 1.1: ICN functionalities in a ITS scenario.

to accomplish this process, each NDN node is equipped with three main architectural elements: Content Store (CS), Pending Interest Table (PIT), and Forwarding Information Base (FIB). Specifically, the Content Store (CS) is a memory that caches incoming *Data* packets; the Pending Interest Table (PIT) tracks user requests not yet satisfied; the FIB maps content names and output interface for supporting routing operations.

The example depicted in Fig. 1.2 describes the NDN mechanism. In the considered scenario, a vehicle communicates with the Control Centre, by expressing an *Interest* for a particular content (e.g. traffic or weather information). When a NDN node receives this *Interest*, it executes the following operations: first of all, the CS is looked up in order to verify if the received request can be satisfied locally; if so, the node sends back the requested *Data*. Otherwise, the PIT is looked up to check if the same *Interest* has been previously forwarded but it is still unsatisfied. In this case, the received *Interest* is discarded, and its arrival face is added to the incoming faces of the matched PIT entry. On the contrary, if a matching entry is not found in the PIT, the FIB is examined through a Longest Prefix Match (LPM) operation, in order to find potential routes to forward the *Interest* through. For what concerns the reception of a *Data* packet, instead, the PIT is the first structure to be checked, because only if there is a matching pending entry, the packet will be processed and sent back to its requester(s); otherwise, it will be discarded.

Publish Subscribe Internet Technolog (PURSUIT)

PURSUIT [9] leverages two communication primitives (i.e., *Publish* and *Subscribe*) and makes use of three logical nodes (Rendezvous Node, Topology

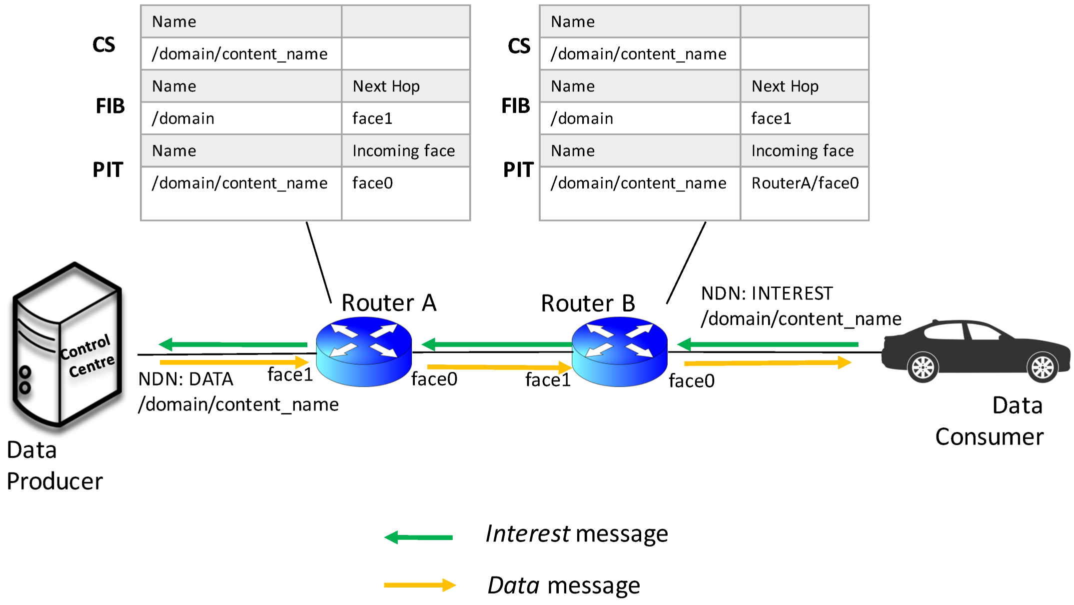


FIGURE 1.2: An example showing NDN functionalities.

Manager and Forwarding Node). A producer advertises a generated content under a given topic, through a *Publish* message. A consumer, willing to subscribe to a specific topic, issues a *Subscribe* message to its local Rendezvous Node. Then, the Rendezvous Network finds the match between the subscription and the content. When this match is resolved, the producer contacts the Topology Manager in order to retrieve the list of Forwarding Nodes through which reach the consumer. Note that this route only provides the unidirectional path. Now, the producer can send updates, until the consumer unsubscribes from the topic by sending an *Unsubscribe* message.

Data-Oriented Network Architecture (DONA)

DONA [7] leverages two communication primitives (i.e., *Find* and *Register*) and a new network entity (i.e., Resolution Handler). A producer expresses a content availability by sending a *Register* message to its local Resolution Handler. This node forwards the packet to peer and uplevel Resolution Handlers, according to local routing configurations. Each Resolution Handler maintains a Registration Table that maps each content name to the next hop Resolution Handler and to the distance from the remote copy. On the other hand, a consumer interested in a content sends a *Find* message to its local Resolution Handler. When a Resolution Handler receives a *Find* message, it checks if there is a corresponding entry in the Registration Table. If so, the packet is forwarded to the next Resolution Handler indicated by the table; otherwise it is sent to the parent Resolution Handler. Moreover, the current node appends the next-hop to the packet field indicating the source address. When the *Find* message reaches the producer, the collected path labels are reversed and exploited to send the content back to the consumer. In this way, a symmetric routing takes place. To retrieve the next content, a new *Find* message has to be issued.

Network of Information (NetInf)
NetInf [8] leverages two communication primitives (i.e., *Publish* and *Get*) and introduces the new network entity known as Content Router. The routing protocol advertizes available content names and populates routing table of Content Routers. A consumer, interested in a specific content, sends a *Get* message with the corresponding name to its local Content Router. This packet propagates hop-by-hop toward the producer or the nearest cache, according to routing information available in Content Routers. Moreover, each node crossed by the *Get* message adds the name of the next-hop Content Router to the packet label stack. Then, the content is delivered within a *Data* packet that follows the reverse path of the *Get* message, obtained by reversing the router names in the label stack.

COntent Mediator architecture for content-aware nETworks (COMET)
COMET [11] leverages two communication primitives (i.e., *Publish* and *Consume*) and two new network entities (i.e., Content Resolution System and Content Aware Forwarding Entity). From one side, the producer announces the availability of a given content through a *Register* message, sent by the producer to its local Content Resolution System. This entity saves a mapping between producer location and content name. Then, it forwards the information in a *Publish* message, upstream in the Autonomous System hierarchy. Each parent Content Resolution System saves a pointer to its child that sent the *Publish* message. From another side, the consumer sends a *Consume* message with the name of the desired content to its local Content Resolution System. Then, this request propagates through the hierarchy, according to routing table. The forwarding process continues until an information about the searched content is found. Moreover, a Content Resolution System that receives a *Consume* message, configures the corresponding Content Aware Forwarding Entity in its local domain. This mechanism prepares the actual delivery of the content, back to the consumer. In fact, when the *Consume* packet reaches the producer, the generated content is sent back to the consumer in a *Data* packet forwarded by Content Aware Forwarding Entities. Then, the next *Consume* packet has to be sent to fetch the next content.

1.3.1 ICN in the IoT

Several works [61]-[64] discussed the benefits of ICN for the IoT. Herein, the advantage of ICN for the IoT are discussed [19][65][66][67][17][68]:

- Information Naming: ICN introduces the decoupling of name and location. An information objects represents the information itself independently of its storage location and physical representation. Therefore, it is possible to directly address heterogeneous IoT contents and services;

- Information Delivery: ICN well suits the inherent information-centric nature of the IoT applications, since the data itself matters and not the producer and its location in the system. A content request can be satisfied by the original information source or by in-network caches. Using this solution, delivery times are reduced and data producers are no more the bottleneck of the network. In addition, multiple users can be interested in the same content. To this end, ICN natively support multicast communications by aggregating request messages for the same information object;

- Mobility: the ICN use of named data facilitates the mobility support because a user only needs to request the desired content to the network. Then, the anycasting and in-network caching functionalities allow the consumer to retrieve the desired content from the most convenient producer/storage point, thus reducing data latency and network traffic. Moreover, ICN can also support a store-carry-and-forward mechanism, by allowing a mobile node to serve as a link (data mule) between disconnected areas. This would allow the communications even under intermittent connectivity;

- Security: ICN natively provides content-based security by securing the content itself (at the packet level), instead of securing the communication channel. Therefore, all the process to establish a secure connection is no more needed. Moreover, ICN supports built-in security because it is receiver-driven: a data can be received only if it is explicitly requested by the receiver. This mechanism can reduce unwanted data transfer, such as spam.

1.3.2 Publish-subscribe Paradigm

As already mentioned, ICN natively provides an explicit separation between contents and locators, identifies data with names (that do not contain any details about the location), handles data request and dissemination through routing-by-name strategies, and allows a simplified management of user mobility [67][69][70]. But, how to efficiently reach a seamless mobility is still an open issue. Works, such as [43][71][72], converge on the necessity of a decoupling of time and space between request resolution and data transfer. Subscribers and publishers do not need to be connected to the network at the same time. They do not usually have information about each other, such as location, number of users interested in a specific content, or number of publishers that are providing the contents. This could be achieved through the publish-subscribe communication model. A publish-subscribe scheme is based on the following principle: the consumer issues a subscription request for a given topic. Then, every time a new content is generated under that topic, the producer (or, more in general, the network) delivers that *Data* to all the subscribed users [24].

The publish-subscribe schema proposed in the literature can be grouped in two main categories, *pull-based* and *push-based* [73][70]. In the first case, the consumer polls the producer to retrieve the next update. In the second case, the consumer establishes a semi-persistent communication path with the producer that sends updates without being solicited.

Aim of this thesis is the design and evaluation of ICN-based publish-subscribe communication schemes for IoT applications, and specifically ITSs.

1.3.3 ICN in the the real world

Before we dive into the work of this thesis, we give a brief overview of the ICN standardisation efforts and its implementations in the real world. Plenty of works have been dedicated to ICN, but they all look like isolated islands, hence, the Internet Engineering Task Force (IETF) started a standardization process [17]-[60].

Three deployment options have been identified [74]:

- clean slate ICN: ICN is supposed to entirety replace the TCP/IP protocol. ICN would lay directly on Layer 2 and all the IP routers would be replaced with routers equipped for instance with NDN Forwarding Daemon (NFD) or PURSUIT forwarding nodes;

- ICN-as-an-Overlay: ICN is an overlay over IP. The ICN overlay can be implemented directly over User Datagram Protocol (UDP) [75]. This approach results in islands of ICN deployments over existing IP-based infrastructure, that are typically connected to each other via ICN/IP tunnels. ICN-as-an-Overlay can be deployed over IP infrastructure in either edge or core networks;

- ICN-as-an-Underlay: ICN would integrate with existing (external) IP-based networks by deploying application layer gateways at appropriate locations [76]. Also this approach results in islands of native ICN deployments which are connected to the rest of the Internet through protocol conversion gateways or proxies.

Beyond the standardization efforts and outside the academia, it is important to remark the Hybrid ICN [77] produced by Cisco, which integrates ICN into the existing IP infrastructure, rather than as an overlay or replacement of IP. A hybrid ICN-IP router processes and forwards both IP packets according to the standard way, and IP packets with an ICN semantic.

Chapter 2

Related work

As discussed in Section 1.3.2, the publish-subscribe schema proposed in the literature can be grouped in two main categories, *pull-based* and *push-based* [73]. In the first case, the consumer polls the producer to retrieve the next update. In the second case, the consumer establishes a semi-persistent communication path with the producer that sends the updates without being solicited. Most of ICN architectures support the publish-subscribe communication schema by natively enabling pull-based or/and push-based mechanisms. A summary of the implemented approaches is reported in Table 3.2. More details are discussed in the following sections [78].

2.1 PURSUIT

PURSUIT natively supports the push-based communication scheme: a consumer issues a *Subscribe* message to receive all updates on a specific topic and an *Unsubscribe* message to interrupt the flow.
On the other hand, to support the pull-based approach, some architectural enhancements are needed. The work presented in [67] suggests to exploit the rendezvous process to retrieve not only the unidirectional path, but the bidirectional ones between producer and consumer. This information allows the subscribed consumer to directly contact the publisher with the classic pull-based approach. Moreover, by using a sliding window mechanism, consecutive requests can be handled.

2.2 DONA

Differently from PURSUIT, DONA inherently supports the pull-based approach and needs specific extensions to implement the push-based one. In particular, since a *Find* message retrieves only one data, consecutive contents can be fetched by using a sliding window mechanism. In this way, the pull-based approach can be supported.
DONA can be extended to manage also the push-based schema. In this context, the work in [7] suggests to add a Time To Live (TTL) field to the *Find* packet. In this way, when the producer receives the message, it caches the packet and sends the update for the specified time. When such a time interval expires, the packet is discarded and a new *Find* packet needs to be sent to refresh the subscription.

TABLE 2.1: Required enhancements to support publish-subscribe in ICN architectures

	Pull-based approach implementation	**Push-based approach implementation**
PURSUIT	Bidirectional path retrieved from the rendezvous process and a sliding window mechanism to handle consecutive *Subscribe* packets	Natively supported by *Subscribe* and *Unsubscribe* messages
DONA	Sliding window mechanism for controlling consecutive *Find* packets	TTL field in the *Find* packet format
NetInf	Sliding window mechanism for controlling consecutive *Get* packets	TTL field in the *Get* packet format
COMET	Sliding window mechanism for controlling consecutive *Consume* packets	TTL field in the *Consume* packet format
NDN	Sliding window mechanism for controlling consecutive *Interest* packets	New type of messages (*Publish* and *Subscribe*) or semipermanent PIT or data within the *Interest* packet

2.3 NetInf

Also NetInf inherently supports the pull-based approach. In fact, a consumer that issues a *Get* message retrieves just one data. In order to receive consecutive updates, a sliding window mechanism could be adopted.
On the other hand, the push-based implementation can be achieved by adding a TTL field to the *Get* packet, such as in DONA.

2.4 COMET

In COMET, the consumer sends a *Consume* message with the desired content name and retrieves the corresponding data. To receive the next content, a new *Consume* packet has to be sent. Consecutive requests can be managed exploiting the sliding window mechanism. Accordingly, COMET supports the pull-based approach.
Instead, the push-based communication schema can be implemented by maintaining active the established multi-hop path. Similar to the previous cases, this goal can be achieved by adding a packet field with a timeout.

2.5 NDN

NDN does not natively support the publish-subscribe communication scheme. But, few extensions were already proposed in the literature. A pull-based implementation, that does not introduce any extension to the baseline NDN architecture, is proposed in [30]. Here, a consumer sends a window of Interest packets in order to receive consecutive updates of the desired content. These requests are forwarded towards the best content location according to FIB entries of intermediate routers. As soon as the consumer receives a Data packet, its window slides to release the next Interest packet.

The work presented in [79], instead, proposes a pull-based approach that adds broker nodes to the original architecture. Producer and consumer enroll by sending an Interest packet to the reference broker node. In particular, the Interest packet sent by the producer includes the name of the content it can generate; the Interest packet sent by the consumer includes the name of the desired content. Then, the broker node confirms the registration through a Data packet. Every time a new content is generated, the broker node notifies the consumer about the content availability and issues the next Interest packet to the producer. Now, the consumer can retrieve the content by using the conventional request-response mechanism.

Some *push-based* approaches are summarized in [18] and deeply investigated in [31], [33] and [34]. Differently from the previous approach, all of these solutions require some enhancements to the baseline NDN communication primitives. [33] suggests to include contents within *Interest* messages sent by the producer towards subscribed consumers. This mechanism is also applied in [34], but a dummy *Data* packet for acknowledgement is introduced in order to preserve the 1-to-1 matching between *Interest* and *Data*. An interesting approach comes from [31][34], where semi-permanent *Interest* packets are used. According to NDN specifications, these kind of *Interest* packets are stored in PIT entries of intermediate routers. However, they are not deleted if one or more contents are sent back to the consumer. Instead, they remain in PIT tables till the expiration of a timeout. Subscriptions are periodically refreshed by issuing new semi-permanent *Interest* packets related to the considered topic.

Push-based publish-subscribe schema can also be implemented by adopting a new set of messages and architectural elements [80]. For instance, [32] presents the COPSS (Content-Oriented Pub/Sub System) architecture, which introduces a new network element (i.e., the Rendezvous Point), two new messages (i.e., subscribe and publish), and a new table in intermediate routers (i.e., Subscription Table). To make a subscription, the consumer sends a subscribe message to a given Rendezvous Point. Multiple Rendezvous Points can be deployed in the network, for handling subscriptions related to different groups of topics. Subscribe messages are routed to the most suitable Rendezvous Point, according to information stored in FIBs of intermediate routers. During the delivery of subscribe messages, intermediate routers also populates their Subscription Table. This table is used to track this new control message in the same way the Pending Interest Table handles *Interest* messages. When a new content is generated, the producer sends it within a publish message to the reference Rendezvous Point. The received message is forwarded to the subscribers, based on information stored in Subscription Tables. The COPSS architecture was applied to disaster scenarios [81] and vehicular networks [82].

In conclusion, the message sequence charts of the discussed implementations are shown in Figure 2.1.

Among all the contributions mentioned before, only [82] addresses also the mobility. To this end, to support the consumer's mobility, the subscription request is periodically renewed; to support the producer's mobility, the

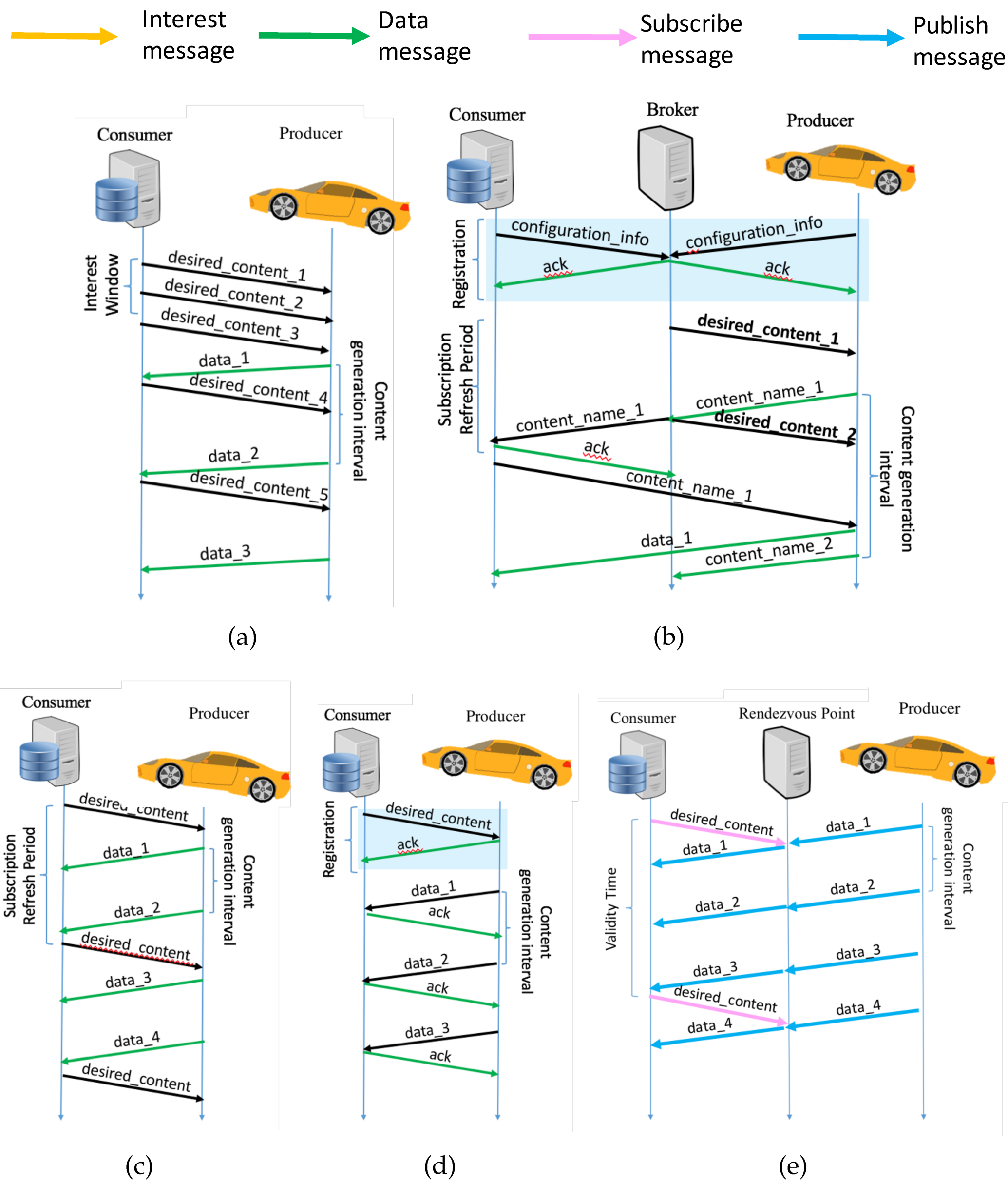


FIGURE 2.1: Message sequence charts for NDN publish-subscribe implementations: (a) pull-based approach with sliding window mechanism; (b) pull-based approach with broker node; (c) push-based approach with semipersistent PIT; (d) push-based approach with data included within the *Interest* packet; (e) push-based approach with Rendezvous Point.

TABLE 2.2: State of the art solutions for publish-subscribe.

	[30]	[18]	[31]	[33]	[34]	[32]	[82]
Pull-based	yes	no	no	no	no	no	no
Push-based	no	yes	yes	yes	yes	yes	yes
Requires extensions to NDN	no	yes	yes	yes	yes	yes	yes
Supports mobility	no	no	no	no	no	no	yes
Provides analytical model	no	no	no	no	yes	no	no

subscription request is sent again if no *Data* are received for a specific amount of time.

To summarize, a big picture of the works investigated in this section is presented in Table 2.2, highlighting design principle (pull or push), the need of an extension of NDN communication primitives, the support for the mobility, as well as analytical models provided. From this table, it clearly emerges that a comprehensive study of publish-subscribe approaches for NDN, which deeply investigates the impact of the mobility, is still missing in the current literature. This lack is going to be addressed in this thesis.

Chapter 3

Publish-subscribe in ICN

In this chapter, new pull-based and push-based publish-subscribe communication schema, able to support user mobility in ICN networks, will be presented. The investigated ICN publish-subscribe approaches can fulfill IoT requirements, such as real-time communication, data-centric service, seamless support of mobile users, and efficient and scalable dissemination process. Moreover, the communication overhead induced by these approaches will be analytically modelled [73][70].

3.1 Designing of publish-subscribe schema in NDN

The reference scenario considered in this contribution is depicted in Figure 3.1. An overlay NDN network[1] is configured on top of network attachment points [17][83][84]. Without loss of generality, it is assumed that producer and consumer are not connected to the same network attachment point. Therefore, the *Data* exchange requires the set up of a multi-hop communication path. Also, producer and consumer could be mobile: while they change the network attachment point, the multi-hop communication paths should be frequently re-configured. The present contribution assumes that such a multi-hop communication path represents the shortest path connecting consumer and producer available within a given network topology. Without loss of generality, the way this path is configured is neglected.

Contents are disseminated according to the publish-subscribe scheme. The producer generates contents for a given topic, namely **ndn://[topic-name]**. Let T_D be the average time interval between the generation of two consecutive contents, belonging to the same topic. Then, each content is uniquely identified with an incremental identifier. The resulting name appears in the form:

ndn://[topic-name]/#id.

Based on these premises, this section (i) provides a formal definition of the communication overhead, (ii) describes some enhancements to baseline *pull-based* and *push-based* approaches required to support the mobility in NDN networks, and (iii) proposes analytical models able to quantify communication overheads due to mobility management.

[1]The same considerations apply for ICN realms that embed NDN functionalities at the network layer.

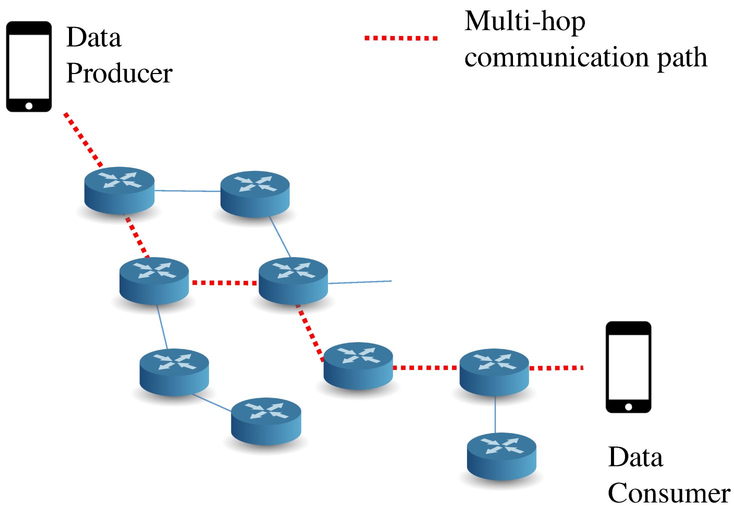


FIGURE 3.1: Reference scenario.

To ease the comprehension of the notions presented in the following, a summary of main symbols is reported in Table 3.1.

3.2 Formal definition of the communication overhead

As anticipated in the Introduction, the implementation of publish-subscribe mechanisms in mobile NDN deployments can generate a non negligible overhead. NDN natively relies on a receiver-driven communication paradigm: a content can be disseminated only in response to an Interest. Note that *Interest* packets cover a key role also in the publish-subscribe mechanism. In fact, in the current literature (see Section 2.5), they act as control messages needed to establish and maintain the multi-hop communication path between consumers and producers. Therefore, they represent the first contribution to the overhead.

A given communication path may remain active until entries stored in PIT tables of intermediate NDN routers expire. When a consumer switches the network attachment point, a new path is created. Path established in the past and not yet expired are simply referred to as *stale paths*. Specifically, routers belonging to stale paths store, for a given transient and in their PIT tables, the request previously sent by the user. Therefore, as soon as the producer generates a new content, it will be disseminated along all the active paths, including the stale ones. Note that stale paths could be partially overlapped with respect to the latest generated one. Hereby, the concept of *stale disjoint links* is introduced to identify the set of links that belong to stale paths and that are not present in the new one. As a consequence, *Data* packets transmitted through stale disjoint links inflates the overhead.

To provide a further insight, Figure 3.2 qualitatively explains the contribution to the communication overhead due to both *Interest* and *Data* packets.

TABLE 3.1: List of mathematical symbols

Symbol	Description
T_D	Time interval between the generation of two consecutive contents
t_i	Time instant in which a communicating entity (i.e., consumer or producer) attaches to the generic *ith* network node
$\Delta\bar{t}^C$	Average cell residence time of the consumer
$\Delta\bar{t}^P$	Average cell residence time of the producer
$O_I^{i,i+1}$	Communication overhead due to control messages during $\Delta t_{i,i+1}$
$O_D^{i,i+1}$	Communication overhead due to data dissemination during $\Delta t_{i,i+1}$
$\bar{O}$	Average communication overhead
W	Window size, adopted in the pull-based approach
T^{RTT}	Round Trip Time
T_{IL}	Interest lifetime
T_O^{pull}	Pending request timeout, adopted in the pull-based approach
d_i	Shortest path between consumer and producer, established at t_i
$\mathcal{A}(t)$	Stale disjoint links in a specific time instant
S_I	Interest message size
S_D	Data message size
T_O^{push}	Pending request timeout, adopted in the push-based approach
R_D	Data packets generation rate
$\bar{v}$	Average speed of the mobile consumer or mobile producer
r	Cell radius
N	Total number of nodes in the topology

The first example refers to the scenario where the consumer is mobile and the producer is stationary. In this case, the consumer sends *Interest* packets to the producer while it is moving across network attachment points. All of these messages provide the first contribution to the communication overhead. When the producer generates a new content, it is delivered across the latest path established between consumer and producer and towards the two stale paths, that in the depicted example are still active. Thus, *Data* packets sent through stale disjoint links give the second contribution to the communication overhead. In the second example, the consumer is stationary and the producer is mobile. Differently from the previous case, now the producer may deliver *Data* packet only through the latest path established with the consumer. As a result, the communication overhead can only be generated by the exchange of *Interest* packets. The scenario considering both consumer and producer mobility embraces all the events described for the first two examples. Here, the communication overhead is generated by the exchange of *Interest* packets (sent from the consumer to the producer) and by *Data* packets (sent through stale disjoint links).

Now, to provide a formal definition of the communication overhead, it is important to introduce the following details. Let t_i be the time instant in which a communicating entity (i.e., a consumer or a producer) attaches to the generic *ith* network attachment point. Then, the *cell residence time*, $\Delta t_{i,i+1}$, represents the amount of time in which the consumer or the producer remains connected to the *ith* network attachment point before jumping to the next one. Moreover, let $O_I^{i,i+1}$ be the total amount of bits due to *Interest* packets transmitted, during $\Delta t_{i,i+1}$, across the links belonging to multi-hop paths connecting consumer and producer. Also, let $O_D^{i,i+1}$ be the total amount of bits due to *Data* packet transmitted across the stale disjoint links, during $\Delta t_{i,i+1}$. The *average communication overhead* that can be produced by a publish-subscribe communication scheme during a unit of time, i.e., $\bar{O}$, can be formally defined as:

$$\bar{O} = \frac{E[O_I^{i,i+1}] + E[O_D^{i,i+1}]}{E[\Delta t_{i,i+1}]}, \tag{3.1}$$

where the $E[x]$ operator returns the expectation of the random variable x. Closed-form expressions for the average communication overhead related to both pull-based and push-based approaches in different mobility scenarios are formulated in the following subsections. Nevertheless, a summary of conceived models is reported in Table 3.2.

3.3 Pull-based approach

The pull-based publish-subscribe communication scheme was initially described in [30]. To support the mobility, it should be extended as described below:

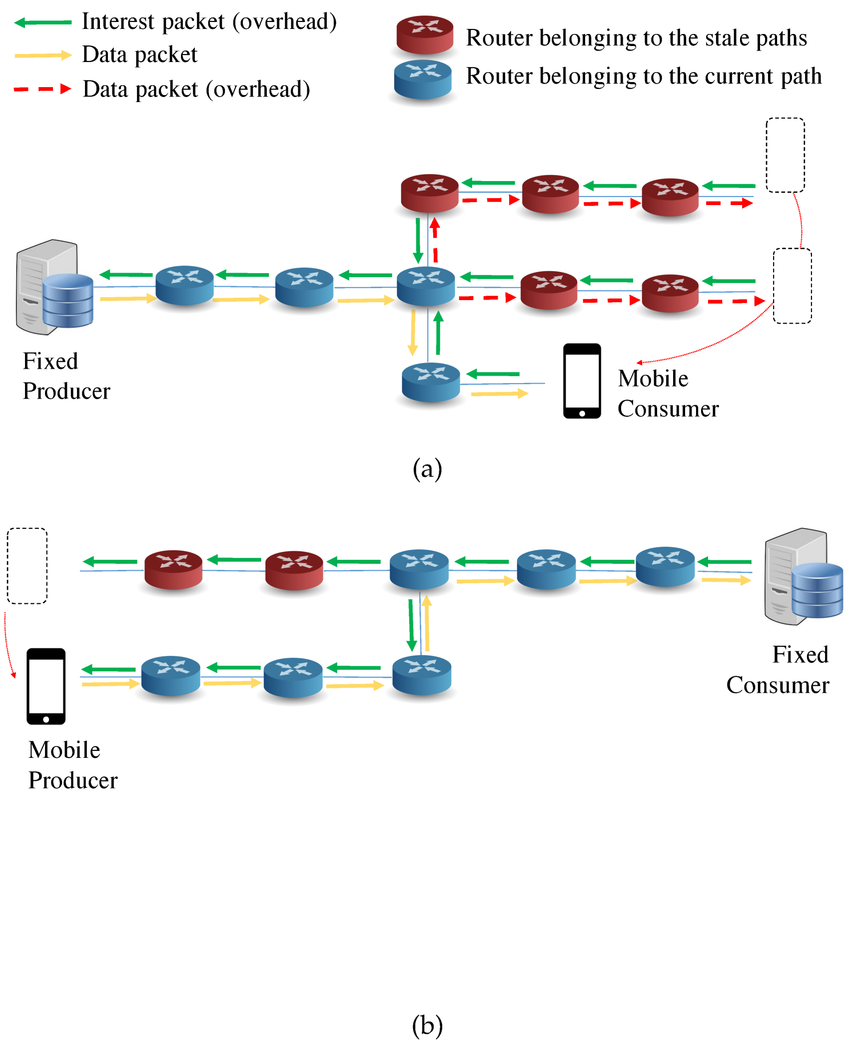


FIGURE 3.2: Examples showing the communication overhead in a scenario with (a) stationary producer and mobile consumer and with (b) mobile producer and stationary consumer.

TABLE 3.2: Summary of analytical models

	Stationary producer, mobile consumer	
	Pull-based	Push-based
$E[O_I^{i,i+1}]$	$\frac{\bar{d}S_I}{\Delta\bar{t}^C}\left(1+\frac{\Delta\bar{t}^C}{T_D}\right)$	$\bar{d}S_I \frac{1}{1-e^{-\frac{1}{\Delta\bar{t}^C}T_O^{push}}}$
$E[O_D^{i,i+1}]$	$\bar{A}S_D$	$\bar{A}R_D\left(\frac{T_O^{push}}{1-e^{-\frac{1}{\Delta\bar{t}^C}T_O^{push}}}-\Delta\bar{t}^C\right)$
$\bar{O}$	$\frac{\bar{d}S_I}{\Delta\bar{t}^C}\left(1+\frac{\Delta\bar{t}^C}{T_D}\right)+\frac{\bar{A}S_D}{\Delta\bar{t}^C}$	$\frac{\bar{d}S_I\frac{1}{1-e^{-\frac{1}{\Delta\bar{t}^C}T_O^{push}}}+\bar{A}R_D\left(\frac{T_O^{push}}{1-e^{-\frac{1}{\Delta\bar{t}^C}T_O^{push}}}-\Delta\bar{t}^C\right)}{\Delta\bar{t}^C}$
	Mobile producer, stationary consumer	
	Pull-based	Push-based
$E[O_I^{i,i+1}]$	$\bar{d}S_I\frac{\Delta\bar{t}^P}{T_D}$	$\bar{d}S_I\frac{\Delta\bar{t}^P}{T_O^{push}}$
$E[O_D^{i,i+1}]$	0	0
$\bar{O}$	$\bar{d}S_I\frac{1}{T_D}$	$\bar{d}S_I\frac{1}{T_O^{push}}$
	Mobile producer, mobile consumer	
	Pull-based	Push-based
$E[O_I^{i,i+1}]$	$\frac{\bar{d}S_I}{\Delta\bar{t}^C}\left(1+\frac{\Delta\bar{t}^C}{T_D}\right)$	$\bar{d}S_I \frac{1}{1-e^{-\frac{1}{\Delta\bar{t}^C}T_O^{push}}}$
$E[O_D^{i,i+1}]$	$\bar{A}S_D$	$\bar{A}R_D\left(\frac{T_O^{push}}{1-e^{-\frac{1}{\Delta\bar{t}^C}T_O^{push}}}-\Delta\bar{t}^C\right)$
$\bar{O}$	$\frac{\bar{d}S_I}{\Delta\bar{t}^C}\left(1+\frac{\Delta\bar{t}^C}{T_D}\right)+\frac{\bar{A}S_D}{\Delta\bar{t}^C}$	$\frac{\bar{d}S_I\frac{1}{1-e^{-\frac{1}{\Delta\bar{t}^C}T_O^{push}}}+\bar{A}R_D\left(\frac{T_O^{push}}{1-e^{-\frac{1}{\Delta\bar{t}^C}T_O^{push}}}-\Delta\bar{t}^C\right)}{\Delta\bar{t}^C}$

- the consumer issues contents' requests (i.e., *Interest* packets) according to a sliding window mechanism: a burst of W requests is sent at the beginning; then, every time a new content is received, a new request is released;

- to optimize the flow control, the window size should be properly configured as a function of both Round Trip Time (RTT) and T_D:

$$W = \left\lceil \frac{T^{RTT}}{T_D} \right\rceil, \tag{3.2}$$

  where the $\lceil x \rceil$ operators returns the nearest integer greater than or equal to x. In general, the flow control acts as a receiver-driven Go-Back-N automatic repeat-request (ARQ)[2]. Most of IoT applications generate data sporadically and $\frac{T^{RTT}}{T_D} < 1$ (see section 4.1.2 for more details). For this reason, we can simplify Eq. (3.2) by setting $W = 1$. Therefore, the flow control mechanism becomes a receiver-driven stop-and-wait ARQ;

- in NDN, the *Interest lifetime* represents the timeout used by network nodes to remove stale Interests from the PITs. In order to avoid that PIT entries are deleted before the reception of a *Data* packet, the *Interest lifetime* (T_{IL}) is set as in the following:

$$T_{IL} \geq T^{RTT} + (T_D W - T^{RTT}). \tag{3.3}$$

  Now, considering that in our case $W = 1$, Eq. (3.3) can be simply rewritten as $T_{IL} = T_D$.

- the consumer assigns a timeout T_O^{pull} to each pending request. When the timeout expires, the same request is sent (again) to the producer. In order to reduce the polling frequency, the pending request timeout T_O^{pull} is set equal to T_{IL}. Indeed, is it possible to set $T_O^{pull} = T_{IL} = T_D$. The polling strategy natively permits the establishment of a new multi-hop communication path in the case of producer mobility.

  For the sake of simplicity, we assume that the update of FIBs tables in intermediate NDN routers is a task of the network [69][85], thus becoming out-of-scope of the publish-subscribe approach;

- if the consumer changes the network attachment point, the latest W requests not yet satisfied are sent again to the producer. In this way, a new path between consumer and producer is created. According to [86] and [87], this strategy supports consumer mobility.

[2]The proposed mechanism reminds Go-Back-N ARQ and does not want to substitute it. Moreover, NDN does not have a separate transport layer and the network layer lays over the link layer. Transport layer's functions are moved to applications and to the strategy component in the forwarding plane [96].

First use case: stationary producer, mobile consumer.

Theorem 1. *Let $\bar{d}$, S_I, $\bar{\Delta t}^C$, T_D, $\bar{\mathcal{A}}$, S_D be the average shortest path, the Interest packet size, the average cell residence time of the consumer, the time interval between the generation of two consecutive contents, the average number of stale disjoint links, and the Data packet size, respectively. The average communication overhead generated by the pull-based approach in the case the producer is stationary and the consumer is mobile is equal to:*

$$\bar{O} = \frac{\bar{d}S_I}{\bar{\Delta t}^C}\left(1 + \frac{\bar{\Delta t}^C}{T_D}\right) + \frac{\bar{\mathcal{A}}S_D}{\bar{\Delta t}^C}. \tag{3.4}$$

Proof. At the beginning of a given cell residence time $\Delta t^C_{i,i+1}$, the mobile consumer polls the stationary producer by sending $W = 1$ *Interest* packet. Then, a new *Interest* packet is issued when a new content is received or the timeout, T_D, assigned to the pending request, expires. This is done till t_{i+1}. During $\Delta t^C_{i,i+1}$, the average number of *Interest* packets sent by the consumer is approximately equal to $\frac{\Delta t^C_{i,i+1}}{T_D}$. Therefore, the resulting communication overhead due to control messages is equal to:

$$O_I^{i,i+1} = d_i S_I + \frac{\Delta t^C_{i,i+1}}{T_D} d_i S_I, \tag{3.5}$$

where d_i and S_I are the shortest path between consumer and producer, established at t_i, and the *Interest* packet size, respectively.

Let $\mathcal{A}(t)$ be the number of stale disjoint links, active when the producer generates a new content. Accordingly, the communication overhead due to extra data dissemination (i.e., *Data* packets sent across stale disjoint links) can be expressed as:

$$O_D^{i,i+1} = \mathcal{A}(t) S_D, \tag{3.6}$$

where S_D represents the size of the *Data* packet.

Now, by substituting Eq. (3.5) and Eq. (3.6) in Eq. (3.1), the average overhead generated by the pull-based approach in the case the consumer is mobile and the producer is stationary becomes:

$$\begin{aligned}\bar{O} &= \frac{E\left[d_i S_I + \frac{\Delta t^C_{i,i+1}}{T_D} d_i S_I\right] + E\left[\mathcal{A}(t) S_D\right]}{E[\Delta t^C_{i,i+1}]} \\ &= \frac{E[d_i]S_I}{E[\Delta t^C_{i,i+1}]} + E[d_i]\frac{S_I}{T_D} + \frac{E[\mathcal{A}(t)]S_D}{E[\Delta t^C_{i,i+1}]}\end{aligned} \tag{3.7}$$

By setting $E[d_i] = \bar{d}$, $E[\Delta t^C_{i,i+1}] = \bar{\Delta t}^C$, and $E[\mathcal{A}(t)] = \bar{\mathcal{A}}$, Eq. (3.7) can be rewritten as:

$$\bar{O} = \frac{\bar{d}S_I}{\bar{\Delta t}^C} + \bar{d}\frac{S_I}{T_D} + \frac{\bar{\mathcal{A}}S_D}{\bar{\Delta t}^C} = \frac{\bar{d}S_I}{\bar{\Delta t}^C}\left(1 + \frac{\bar{\Delta t}^C}{T_D}\right) + \frac{\bar{\mathcal{A}}S_D}{\bar{\Delta t}^C}. \tag{3.8}$$

□

Second use case: mobile producer, stationary consumer.

Theorem 2. *Let $\bar{d}$, S_I, and T_D be the average shortest path, the Interest packet size, and the time interval between the generation of two consecutive contents, respectively. The average communication overhead generated by the pull-based approach in the case the consumer is stationary and the producer is mobile is equal to:*

$$\bar{O} = \bar{d}\frac{S_I}{T_D}. \tag{3.9}$$

Proof. Since the consumer does not modify its network attachment point, an *Interest* packet is periodically sent every T_D. Indeed:

$$O_I^{i,i+1} = \frac{\Delta t^P_{i,i+1}}{T_D} d_i S_I. \tag{3.10}$$

Moreover, stale paths are never generated, because the producer may deliver contents it generates only through the latest path established with the consumer. Therefore, the following Equation holds:

$$O_D^{i,i+1} = 0. \tag{3.11}$$

By substituting Eq. (3.10) and Eq. (3.11) in Eq. (3.1), the average overhead generated by the pull-based approach in the case the producer is mobile and the consumer is stationary becomes:

$$\bar{O} = \frac{E\left[\frac{\Delta t^P_{i,i+1}}{T_D} d_i S_I\right]}{E[\Delta t^P_{i,i+1}]} = \frac{\frac{E[\Delta t^P_{i,i+1}]}{T_D} E[d_i] S_I}{E[\Delta t^P_{i,i+1}]} = E[d_i]\frac{S_I}{T_D} \tag{3.12}$$

By setting $E[d_i] = \bar{d}$, it can be rewritten as:

$$\bar{O} = \bar{d}\frac{S_I}{T_D}. \tag{3.13}$$

□

Third use case: mobile producer, mobile consumer.

Theorem 3. *Let $\bar{d}$, S_I, $\bar{\Delta t}^C$, and T_D be the average shortest path, the Interest packet size, the average cell residence time of the consumer and the time interval between the generation of two consecutive contents, respectively. The average communication overhead generated by the pull-based approach in the case both consumer and producer are mobile is equal to:*

$$\bar{O} = \frac{\bar{d}S_I}{\bar{\Delta t}^C}\left(1 + \frac{\bar{\Delta t}^C}{T_D}\right) + \frac{\bar{\mathcal{A}}S_D}{\bar{\Delta t}^C}. \tag{3.14}$$

Proof. At the beginning of a given cell residence time $\Delta t^C_{i,i+1}$, the mobile consumer polls the stationary producer by sending $W = 1$ *Interest* packet. Then, it sends a new *Interest* packet after receiving a new content or the timeout, T_D, assigned to the pending request expires.

During $\Delta t^C_{i,i+1}$, the average number of *Interest* packets sent by the consumer is approximately equal to $\frac{\Delta t^C_{i,i+1}}{T_D}$. Therefore, the resulting communication overhead due to control messages is equal to:

$$O_I^{i,i+1} = d_i S_I + \frac{\Delta t^C_{i,i+1}}{T_D} d_i S_I, \tag{3.15}$$

where d_i and S_I are the shortest path between consumer and producer, established at t_i, and the *Interest* packet size, respectively.

Let $\mathcal{A}(t)$ be the number of stale disjoint links, active when the producer generates a new content. Accordingly, the communication overhead due to *Data* packets sent across stale disjoint links can be expressed as:

$$O_D^{i,i+1} = \mathcal{A}(t) S_D, \tag{3.16}$$

where S_D is the size of the *Data* packet.

It is important to note that, differently from the first use case, the variable $\mathcal{A}(t)$ is influenced by both consumer mobility and producer mobility. Specifically, when the consumer changes the network attachment point, a new multi-hop communication path is established and some of the links belonging to the old one are included in $\mathcal{A}(t)$. On the contrary, as soon the producer changes its network attachment point, $\mathcal{A}(t)$ becomes an empty set. In this case, in fact, all the stale disjoint links will do not produce communication overhead anymore: since the producer is not attached to previous stale paths, next *Data* packets will be sent only through the new established path.

Now, by substituting Eq. (3.15) and Eq. (3.16) in Eq. (3.1), the average overhead generated by the pull-based approach in the case both consumer and producer are mobile becomes:

$$\bar{O} = \frac{E\left[d_i S_I + \frac{\Delta t^C_{i,i+1}}{T_D} d_i S_I\right] + E\left[\mathcal{A}(t) S_D\right]}{E[\Delta t^C_{i,i+1}]}$$
$$= \frac{E[d_i] S_I}{E[\Delta t^C_{i,i+1}]} + E[d_i]\frac{S_I}{T_D} + \frac{E[\mathcal{A}(t)] S_D}{E[\Delta t^C_{i,i+1}]} \tag{3.17}$$

By setting $E[d_i] = \bar{d}$, $E[\Delta t^C_{i,i+1}] = \Delta \bar{t}^C$, and $E[\mathcal{A}(t)] = \bar{\mathcal{A}}$, Eq. (3.17) can be rewritten as:

$$\bar{O} = \frac{\bar{d} S_I}{\Delta \bar{t}^C} + \bar{d}\frac{S_I}{T_D} + \frac{\bar{\mathcal{A}} S_D}{\Delta \bar{t}^C} = \frac{\bar{d} S_I}{\Delta \bar{t}^C}\left(1 + \frac{\Delta \bar{t}^C}{T_D}\right) + \frac{\bar{\mathcal{A}} S_D}{\Delta \bar{t}^C}. \tag{3.18}$$

□

3.4 Push-based approach

According to the standard NDN implementation, the PIT entry associated to a request is deleted as soon as a new *Data* packet is sent back to the consumer. Then, any information related to the communication path between consumer and producer is definitively lost. The push-based approach extends the normal behavior of NDN communication primitives by introducing semi-persistent *Interest* packets [34]. Specifically, a semi-persistent *Interest* packet is not erased from the PIT table even if one or more corresponding *Data* packets were already delivered to the consumer. But, it remains in the PIT table until its timeout expires.

To support the mobility, the push-based approach should be implemented as follows:

- the consumer releases a single request through a semi-permanent *Interest* packet;
- the consumer assigns a timeout, T_O^{push}, to the pending request. When that timeout expires, the same request is sent (again) to the producer. T_O^{push} has been set in order to have only one stale path during $\Delta t_{i,i+1}$, as discussed below. Note that the polling strategy permits the establishment of a new multi-hop communication path in the case of producer mobility;
- if the consumer changes the network attachment point, a semi-permanent *Interest* is sent to the producer. In this way, a new path between consumer and producer is created. Thus, the consumer mobility is supported too.

First use case: stationary producer, mobile consumer.

Theorem 4. *Let $\bar{d}$, S_I, $\Delta\bar{t}^C$, R_D, and $\bar{\mathcal{A}}$ be the average shortest path, the Interest packet size, the average cell residence time of the consumer, the contents' generation rate, and the average set of stale disjoint links, respectively. Moreover, let the pending request timeout, T_O^{push}, be set in order to have only one stale path during $\Delta t^C_{i,i+1}$. The average communication overhead generated by the push-based approach in the case the producer is stationary and the consumer is mobile is:*

$$\bar{O} = \frac{\bar{d}S_I \frac{1}{1-e^{-\frac{1}{\Delta\bar{t}^C}T_O^{push}}} + \bar{\mathcal{A}}R_D\left(\frac{T_O^{push}}{1-e^{-\frac{1}{\Delta\bar{t}^C}T_O^{push}}} - \Delta\bar{t}^C\right)}{\Delta\bar{t}^C}. \tag{3.19}$$

Proof. At the beginning of a given cell residence time $\Delta t^C_{i,i+1}$, the mobile consumer polls the stationary producer by sending an *Interest* packet. When the timeout, T_O^{push}, expires, the same request is sent (again) to the producer. This is done till t_{i+1}. Besides the first request, the additional number of *Interest* packets sent by the consumer in $\Delta t^C_{i,i+1}$ is equal to $\left\lfloor \frac{\Delta t^C_{i,i+1}}{T_O^{push}} \right\rfloor$. Therefore, the communication overhead due to control messages, evaluated during the time interval $\Delta t^C_{i,i+1}$, can be expressed as:

$$O_I^{i,i+1} = d_i S_I + d_i S_I \left\lfloor \frac{\Delta t^C_{i,i+1}}{T_O^{push}} \right\rfloor. \tag{3.20}$$

The adoption of semi-permanent *Interest* implies that a stale path may exist for a long period of time. Also, the number of stale disjoint links may change during the cell residence time. The communication overhead due to transmission of *Data* packets across stale disjoint links could be formally expressed as a function of $\Delta t^C_{i,i+1}$, $\mathcal{A}(t)$, the size of each *Data* packet S_D, and contents' generation rate T_D:

$$O_D^{i,i+1} = \mathcal{F}\left(\Delta t^C_{i,i+1}, \mathcal{A}(t), S_D, T_D\right). \tag{3.21}$$

From Eq. (3.21) it emerges that the estimation of the contribution to the overhead due to extra data dissemination is difficult to achieve. Nevertheless, under some specific assumptions on T_O^{push}, a valid approximation exists.

In this contribution, it is assumed that T_O^{push} is set in order to have only one stale path during $\Delta t^C_{i,i+1}$. According to [88] and [89], the cell residence time can be modeled through an exponential distribution with parameter $1/\Delta\bar{t}^C > 0$:

$$f_{\Delta t^C}(t) = \frac{1}{\Delta\bar{t}^C} e^{-\frac{1}{\Delta\bar{t}^C}t}. \tag{3.22}$$

Thus, in order to have only one stale path (which represents our design assumption), the relation:

$$T_O^{push} < \Delta t_{i,i+1}^C + \Delta t_{i+1,i+2}^C, \tag{3.23}$$

should be valid with a high probability (i.e., $> 99\%$)

Let ϵ be a very small number (i.e., 10^{-2}), Eq. (3.23) is satisfied if:

$$\begin{cases} P(T_O^{push} < \Delta t_{i,i+1}^C + \Delta t_{i+1,i+2}^C) = 1 - P(\Delta t_{i,i+1}^C + \Delta t_{i+1,i+2}^C < T_O^{push}) \\ P(\Delta t_{i,i+1}^C + \Delta t_{i+1,i+2}^C < T_O^{push}) < \epsilon. \end{cases} \tag{3.24}$$

Since the sum of two random variables, characterized by exponential distribution and same average value, is a Gamma distribution[3], the second equation of (3.24) can be rewritten as:

$$P(\Delta t_{i,i+1}^C + \Delta t_{i+1,i+2}^C < T_O^{push}) = \int_0^{T_O^{push}} \left(\frac{1}{\Delta \bar{t}^C}\right)^2 t e^{-\frac{1}{\Delta \bar{t}^C} t} dt < \epsilon, \tag{3.25}$$

that, by integrating by part, becomes:

$$1 - e^{-\frac{1}{\Delta \bar{t}^C} T_O^{push}} \left(\frac{1}{\Delta \bar{t}^C} T_O^{push} + 1\right) < \epsilon. \tag{3.26}$$

Indeed, the values of T_O^{push} that satisfy the initial assumption can be calculated by solving Eq. (3.26), as deeply commented in section 4.1.2.

Under these assumptions, the communication overhead due to the *Data* packets sent through the set of stale disjoint links, $\mathcal{A}(t)$, during the time interval they are still active, that is $T_O^{push} - \Delta t_{i,i+1}^C mod\, T_O^{push} = T_O^{push} - \Delta t_{i,i+1}^C + T_O^{push} \left\lfloor \frac{\Delta t_{i,i+1}^C}{T_O^{push}} \right\rfloor$, is:

$$O_D^{i,i+1} = \left(T_O^{push} - \Delta t_{i,i+1} + T_O^{push} \left\lfloor \frac{\Delta t_{i,i+1}^C}{T_O^{push}} \right\rfloor \right) \mathcal{A}(t) R_D, \tag{3.27}$$

where R_D is the contents' generation rate.

By substituting Eq. (3.20) and Eq. (3.27) in Eq. (3.1), the average communication overhead generated by the push-based approach in the case the

[3] $Gamma\left(2, \frac{1}{\Delta \bar{t}^C}\right) = \left(\frac{1}{\Delta \bar{t}^C}\right)^2 t e^{-\frac{1}{\Delta \bar{t}^C} t}$

producer is stationary and the consumer is mobile becomes:

$$\bar{O} = \frac{E\left[d_i S_I + d_i S_I \left\lfloor \frac{\Delta t^C_{i,i+1}}{T_O^{push}} \right\rfloor\right]}{E[\Delta t^C_{i,i+1}]} + \frac{E\left[\mathcal{A}(t) R_D \left(T_O^{push} - \Delta t_{i,i+1} + T_O^{push} \left\lfloor \frac{\Delta t^C_{i,i+1}}{T_O^{push}} \right\rfloor\right)\right]}{E[\Delta t^C_{i,i+1}]} = \frac{\bar{d} S_I \left(1 + E\left[\left\lfloor \frac{\Delta t^C_{i,i+1}}{T_O^{push}} \right\rfloor\right]\right)}{\bar{\Delta t}^C} + \frac{\bar{A} R_D \left(T_O^{push} - \bar{\Delta t}^C + T_O^{push} E\left[\left\lfloor \frac{\Delta t^C_{i,i+1}}{T_O^{push}} \right\rfloor\right]\right)}{\bar{\Delta t}^C}. \tag{3.28}$$

Now, to solve the expected value of a floor function, let $a = \left\lfloor \frac{\Delta t^C_{i,i+1}}{T_O^{push}} \right\rfloor$ with $a \in \mathbb{Z}$ and let it solve:

$$E[a] = \sum_{a=0}^{\inf} a p_{\Delta t^C}(a). \tag{3.29}$$

From the floor function definition, it results that $a \leq \frac{\Delta t^C_{i,i+1}}{T_O^{push}} < a + 1$, or $aT_O^{push} \leq \Delta t^C_{i,i+1} < (a+1)T_O^{push}$. The probability associated to a is:

$$\begin{aligned} p_{\Delta t^C}(a) &= Pr[aT_O^{push} \leq \Delta t^C_{i,i+1} < (a+1)T_O^{push}] \\ &= \int_{aT_O^{push}}^{(a+1)T_O^{push}} f_{\Delta t}(t) dt = \int_{aT_O^{push}}^{(a+1)T_O^{push}} \frac{1}{\bar{\Delta t}} e^{-\frac{1}{\bar{\Delta t}^C} t} dt \\ &= e^{\frac{1}{\bar{\Delta t}^C} T_O^{push} a} (1 - e^{\frac{1}{\bar{\Delta t}^C} T_O^{push}}) \end{aligned} \tag{3.30}$$

By substituting Eq. (3.30) in Eq. (3.29), it results that:

$$\begin{aligned} E[a] &= (1 - e^{\frac{1}{\bar{\Delta t}^C} T_O^{push}}) \sum_{a=0}^{\inf} a e^{\frac{1}{\bar{\Delta t}^C} T_O^{push} a} \\ &= (1 - e^{\frac{1}{\bar{\Delta t}^C} T_O^{push}}) \frac{e^{\frac{1}{\bar{\Delta t}^C} T_O^{push}}}{(e^{\frac{1}{\bar{\Delta t}^C} T_O^{push}} - 1)^2} \\ &= -\frac{e^{\frac{1}{\bar{\Delta t}^C} T_O^{push}}}{e^{\frac{1}{\bar{\Delta t}^C} T_O^{push}} - 1}. \end{aligned} \tag{3.31}$$

To conclude, by substituting Eq.(3.31) in Eq. (3.28), the average communication overhead becomes:

$$\bar{O} = \frac{\bar{d}S_I \left(1 - \frac{e^{\frac{1}{\Delta \bar{t}^C} T_O^{push}}}{e^{\frac{1}{\Delta \bar{t}^C} T_O^{push}} - 1}\right) + \bar{A}R_D \left(T_O^{push} - \Delta \bar{t}^C - T_O^{push} \frac{e^{\frac{1}{\Delta \bar{t}^C} T_O^{push}}}{e^{\frac{1}{\Delta \bar{t}^C} T_O^{push}} - 1}\right)}{\Delta \bar{t}^C}$$

$$= \frac{\bar{d}S_I \frac{1}{1 - e^{\frac{1}{\Delta \bar{t}^C} T_O^{push}}} + \bar{A}R_D \left(\frac{T_O^{push}}{1 - e^{\frac{1}{\Delta \bar{t}^C} T_O^{push}}} - \Delta \bar{t}^C\right)}{\Delta \bar{t}^C} \tag{3.32}$$

□

Second use case: mobile producer, stationary consumer.

Theorem 5. *Let $\bar{d}$, S_I, and T_O^{push}, be the average shortest path, the Interest packet size, and the pending request timeout, respectively. The average communication overhead generated by the push-based approach in the case the producer is mobile and the consumer is stationary is:*

$$\bar{O} = \frac{\bar{d}S_I}{T_O^{push}}. \tag{3.33}$$

Proof. Since the consumer does not modify its network attachment point, an *Interest* packet is periodically sent every T_O^{push}. Therefore, the contribution to the communication overhead due to *Interest* packets can be expressed as:

$$O_I^{i,i+1} = d_i S_I \frac{\Delta t_{i,i+1}^P}{T_O^{push}}. \tag{3.34}$$

During the cell residence time $\Delta t_{i,i+1}^P$, the producer may deliver contents only through the latest multi-hop path established with the consumer. Hence:

$$O_D^{i,i+1} = 0. \tag{3.35}$$

Therefore, by substituting Eq. (3.34) and Eq. (3.35) in Eq. (3.1), the average communication overhead becomes:

$$\bar{O} = \frac{E\left[d_i S_I \frac{\Delta t_{i,i+1}^P}{T_O^{push}}\right]}{E[\Delta t_{i,i+1}^P]} = \frac{E[d_i] S_I \frac{E[\Delta t_{i,i+1}^P]}{T_O^{push}}}{E[\Delta t_{i,i+1}^P]} = E[d_i] \frac{S_I}{T_O^{push}}. \tag{3.36}$$

By setting $E[d_i] = \bar{d}$, Eq. (3.36) can be rewritten as:

$$\bar{O} = \bar{d} \frac{S_I}{T_O^{push}}. \tag{3.37}$$

□

Third use case: mobile producer, mobile consumer.

Theorem 6. *Let $\bar{d}$, S_I, $\bar{\Delta t^C}$, R_D, and $\bar{\mathcal{A}}$ be the average shortest path, the Interest packet size, the average cell residence time of the consumer, the contents' generation rate, and the average set of stale disjoint links, respectively. Moreover, let the pending request timeout, T_O^{push}, be set in order to have up to one stale path during $\Delta t^C_{i,i+1}$. The average communication overhead generated by the push-based approach when both producer and consumer are mobile is:*

$$\bar{O} = \frac{\bar{d}S_I \frac{1}{1-e^{\frac{1}{\bar{\Delta t^C}}T_O^{push}}} + \bar{\mathcal{A}}R_D\left(\frac{T_O^{push}}{1-e^{\frac{1}{\bar{\Delta t^C}}T_O^{push}}} - \bar{\Delta t^C}\right)}{\bar{\Delta t^C}}. \tag{3.38}$$

Proof. At the beginning of a given cell residence time $\Delta t^C_{i,i+1}$, the mobile consumer polls the stationary producer by sending an *Interest* packet. After the expiration of the timeout, T_O^{push}, the same request is sent (again) to the producer. This is done till t_{i+1}. Besides the first request, the additional number of *Interest* packets sent by the consumer in $\Delta t^C_{i,i+1}$ is equal to $\left\lfloor \frac{\Delta t^C_{i,i+1}}{T_O^{push}} \right\rfloor$. Therefore, the communication overhead due to control messages, evaluated during the time interval $\Delta t^C_{i,i+1}$, can be formulated as:

$$O_I^{i,i+1} = d_i S_I + d_i S_I \left\lfloor \frac{\Delta t^C_{i,i+1}}{T_O^{push}} \right\rfloor. \tag{3.39}$$

Similarly to the first use case, also in this context the number of stale disjoint links, $\mathcal{A}(t)$, may produce overhead during the time interval, whose upper bound is set to $T_O^{push} - \Delta t^C_{i,i+1} mod\, T_O^{push} = T_O^{push} - \Delta t^C_{i,i+1} + T_O^{push} \left\lfloor \frac{\Delta t^C_{i,i+1}}{T_O^{push}} \right\rfloor$. Of course, T_O^{push} is properly set in order to have up to one stale path during $\Delta t^C_{i,i+1}$. Thus, the communication overhead generated by *Data* packet sent through stale disjoint links is equal to:

$$O_D^{i,i+1} = \left(T_O^{push} - \Delta t^C_{i,i+1} + T_O^{push} \left\lfloor \frac{\Delta t^C_{i,i+1}}{T_O^{push}} \right\rfloor \right) \mathcal{A}(t) R_D, \tag{3.40}$$

where R_D is the contents' generation rate.

As expected, the variable $\mathcal{A}(t)$ is influenced by both consumer mobility and producer mobility. Specifically, when the consumer changes the network attachment point, a new multi-hop communication path is established and some of the links belonging to the old one are included in $\mathcal{A}(t)$. On the contrary, as soon the producer changes its network attachment point, $\mathcal{A}(t)$ becomes an empty set. In this case, in fact, all the stale disjoint links will do not produce communication overhead anymore: since the producer is not attached to previous stale paths, next *Data* packets will be only sent through the new established path.

Indeed, by following the same mathematical procedure already presented for the first use case, it is possible to conclude that the average communication overhead is equal to:

$$\begin{aligned}\bar{O} &= \frac{\bar{d}S_I\left(1-\frac{e^{\frac{1}{\bar{\Delta t}^C}T_O^{push}}}{e^{\frac{1}{\bar{\Delta t}^C}T_O^{push}}-1}\right)+\bar{\mathcal{A}}R_D\left(T_O^{push}-\bar{\Delta t}^C-T_O^{push}\frac{e^{\frac{1}{\bar{\Delta t}^C}T_O^{push}}}{e^{\frac{1}{\bar{\Delta t}^C}T_O^{push}}-1}\right)}{\bar{\Delta t}^C}\\ &= \frac{\bar{d}S_I\frac{1}{1-e^{\frac{1}{\bar{\Delta t}^C}T_O^{push}}}+\bar{\mathcal{A}}R_D\left(\frac{T_O^{push}}{1-e^{\frac{1}{\bar{\Delta t}^C}T_O^{push}}}-\bar{\Delta t}^C\right)}{\bar{\Delta t}^C}\end{aligned} \tag{3.41}$$

□

Chapter 4

Publish-subscribe Models Performance Evaluation

This chapter investigates the pull-based and push-based mechanisms described in Chapter 3 and validates related analytical models through computer simulations. To this end, the conducted study considers well-known benchmark network topologies, real IoT monitoring services, and standardized settings for urban and rural environments.

4.1 Scenario and Parameter Settings

4.1.1 Network attachment points and user speed

A number of network attachment points are distributed on a geographical area of 10^6 m^2. They offer wireless connectivity to a mobile user in a coverage area having an average cell radius equal to r. Each attachment point is a NDN node and all of them form an overlay NDN network [17][83][84]. In an ITS scenario, each network attachment point, can represent a Road Side Unit. Usually, vehicle and remote Control Centre are not directly connected. In order to have a successful data delivery, a multi-hop communication path has to be established between them. During the time, the vehicle frequently changes the network attachment point. As a consequence, user mobility implies that the multi-hop communication path established between vehicle and remote Control Centre has to be re-established.

To jointly consider both urban and rural environments, cell radius and user speed are set as reported in Table 4.1 [90].

TABLE 4.1: Cell radius and user speed for both rural and urban scenarios.

	Urban			**Rural**		
Cell radius, r [m]	50	100	150	300	650	1000
Resulting average node density [$nodes/km^2$]	10000	2500	1111	227	61	25
Average user speed, $\bar{v}$ [km/h]	3, 30, 50	3, 30, 50	3, 30, 50	30, 50, 120	30, 50, 120	30, 50, 120

4.1.2 Setting of timers

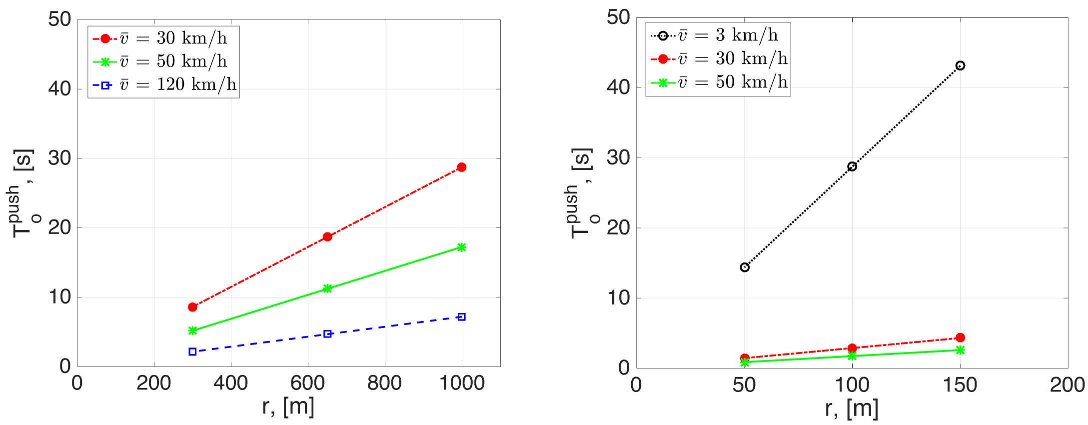
Many time-related parameters should be properly set.

With reference to the average communication overhead related to the push-based approach, section 3.4 already anticipated that the values of T_O^{push} that satisfy the assumption to have only one stale path during a cell residence time can be calculated by solving Eq. (3.26). Now, according to [91], the average cell residence time can be expressed as:

$$\bar{\Delta t}^C = \bar{\Delta t}^P = \frac{\pi r}{2\bar{v}}, \tag{4.1}$$

where r and $\bar{v}$ are the cell radius and the speed of the mobile user (i.e, the consumer or the producer), respectively.

Therefore, by setting $\epsilon = 10^{-2}$, the resulting T_O^{push} values are shown in Figure 4.1. These values are taken into account during computer simulations.



(a) (b)

FIGURE 4.1: T_O^{push} values evaluated for the (a) rural scenario and (b) urban scenario.

Finally, typical T^{RTT} values are of the order of millisecond [92]. Therefore, since $T^{RTT} \ll T_D$, the sliding window size W has been set to 1 in Eq. (3.2).

4.1.3 IoT services and packet size

The conducted study considers realistic IoT applications, as described in [93]:

- Smart Meter (SM): electronic device records energy consumption, allowing the monitoring of power consumption and billing;
- Home Security System (HSS): outdoor and indoor sensors provide motion detection, alarm system, but also gas, water, heating measurement to monitor home status;

- Traffic Sensor (TS): sensors located on vehicles or along routes monitor traffic conditions, travel speed, traffic anomalies, toll highway information or pollution.
- Health Sensor (HS): sensors, such as wearable devices, provide biomedical parameters (heart rate, blood pressure, and so on) to monitor user wellness;

At the application layer, the main simulation settings include (i) the average time interval between the generation of two consecutive contents, T_D, (ii) the application payload, (iii) the *Data* packet size, S_D, (iv) the average content generation rate, $R_D = \frac{S_D}{T_D}$, (v) and the *Interest* packet size, S_I. In line with [93], these parameters are set as reported in Table 4.2. Note that both S_I and S_D are computed by considering the application payload suggested in [93], the average content name of 40 byte [94] and the typical structure of *Interest*, and *Data* packets depicted in Figure 4.2 [95][96].

TABLE 4.2: Parameters related to the considered IoT services.

Application	T_D **[s]**	**App payload [Byte]**	S_D **[bit]**	R_D **[bit/s]**	S_I **[bit]**
SM	9090	2017	18552	2.04	432
HSS	600	20	2576	4.29	432
TS	60	1	2424	40.40	432
HS	60	128	3440	57.33	432

Interest packet format

FIELD	BYTE
Nonce	4
Scope	1
Nack Type	1
InterestLifetime	2
Name	2 + name size
Selectors	2
Options	2

Data packet format

FIELD	BYTE
Name	2 + name size
Content	2 + Application payload
Signature	1 + 256

FIGURE 4.2: *Data* and *Interest* packet formats for NDN.

4.2 Publish-subscribe Models Simulator

As already said, analytical models conceived in Chapter 3 are validated through computer simulations. To this end, two different tools are used: Boston university Representative Internet Topology gEnerator (BRITE) and an ad-hoc simulator.

4.2.1 About BRITE

BRITE [35] is used to generate network topologies that, in line with current literature [97], follow the scale-free model. Indeed, it is adopted to generate the NDN overlay network, as well as to define the distribution of nodes in a pre-defined geographical area and the connection links. In particular, the topology is created step-by-step. At each step, only one node is added to the network and attached to an existing node, properly chosen through a power-law formula [97]. The resulting topology has an average shortest path, $\bar{d}$, approximately equal to [97]: $\bar{d} \approx \log N$, where N is the total number of available nodes. Note that $\bar{d}$ is one of the parameters introduced in Chapter 3 for evaluating the average communication overhead of both pull-based and push-based mechanism. The resulting network topology is then processed by the ad-hoc simulator, as discussed below.

The ad-hoc simulator is written in C++ and provides a system-level model of the publish-subscribe communication scheme based on ICN networking primitives. Conceived as a synchronous simulator, it tracks the status of the network and executes some key operations during the time, useful to calculate (at the end) the communication overhead based on a list of parameter settings.

The tool receives in input the network topology generated with BRITE, the specific approach used for implementing the publish-subscribe communication scheme, the speed of both producer and consumer, and the parameters related to the application layer.

At the beginning of each simulation, the initial position of both producer and consumer is randomly chosen within the geographical area of $10^6 m^2$. Indeed, the tool identifies the network attachment point for both producer and consumer according to the position-based handover algorithm: the user attaches to the nearest node belonging to the modeled network topology.

From this moment on, five parallel processes are executed over the time. They include: *Mobility Manager*, *Handover Manager*, *Paths Manager*, *Interest Generation Process*, and *Data Generation Process*. The algorithm used for the pull-based and push-based approach is shown in Figure 4.3 and Figure 4.4, respectively.

The *Mobility Manager* periodically updates the position of both producer and consumer. This is done according to the well-known random-walk mobility model. Every time a new position is chosen, the tool triggers the execution of functionalities offered by *Handover Manager* and *Paths Manager*. The *Handover Manager* implements the position-based handover algorithm. Thus, it verifies if the current user position requires a new network attachment point. In the affermative case, the *Paths Manager* establishes a new shortest path between consumer and producer and updates the list of disjoint links still available in the network. Note that disjoint links will be deleted form the list handled by the *Paths Manager* when they are crossed by a *Data* packet (for the pull-based approach) or theit assigned lifetime expires (for the push-based approach).

The *Interest Generation Process* runs at the consumer side. It is in charge of sending the control messages (i.e., the *Interest* packets) related to subscription requests. As deeply illustrated in section 3, pull-based and push-based approaches implement their own mechanisms. In the first case, a new control message is sent when a new content is received by the consumer, when a new attachment point is selected for the consumer, or when the timeout T_O^{pull} expires (see section 3.2 for more details). In the second case, instead, a new control message is sent when a new attachment point is selected for the consumer, or when the timeout T_O^{push} expires (see section 3.3 for more details). When the *Interest Generation Process* issues a new control message, the *Paths Manager* provides the list of links through which sending the packet and reports these information in a log file. These details will be processed at the end of the simulation for calculating the first contribution of the communication overhead.

Finally, the *Data Generation Process* creates a new content every an average amount of time equal to T_D. Every time a new content is generated, the *Paths Manager* provides the list of links through which sending the packet and reports these information in a log file. These links include also the stale ones. Also in this case, these details will be processed at the end of the simulation for calculating the secondo contribution of the communication overhead.

The code of the developed ad-hoc simulator is publicly released as an open source project, under the GNU General Public License v03. It can be freely downloaded from the link *telematics.poliba.it/pubsubmodel*.

4.2.2 Towards the validation of analytical models

The *analytical* average communication overhead is provided by analytical models, as described in section Chapter 3. To this end, the Equations reported in Table 3.2 are solved by considering input parameters, the average shortest path given by $\bar{d} = \log N$, and the average number of stale disjoint links obtained as described before.

A bunch of computer simulations are executed to estimate the *simulated* average communication overhead. Specifically, for each combination of parameters settings, 300 different network realizations are simulated. Moreover, for each realization, BRITE is used to generate a new network topology and the ad-hoc simulator is used to calculate the communication overhead. Then, all the obtained results are processed for estimating the *simulated* average communication overhead. Note also that the average number of stale disjoint links $\bar{\mathcal{A}}$, used in Eq. (3.4), Eq. (3.19), Eq. (3.14) and Eq. (3.38), is estimated from simulations. This value is used for estimating the *analytical* average communication overhead, as described below.

Finally, in order to evaluate the accuracy of analytical models, *simulation* and *analytical* average communication overheads are compared through the absolute relative error (see section 4.3.1 for more details).

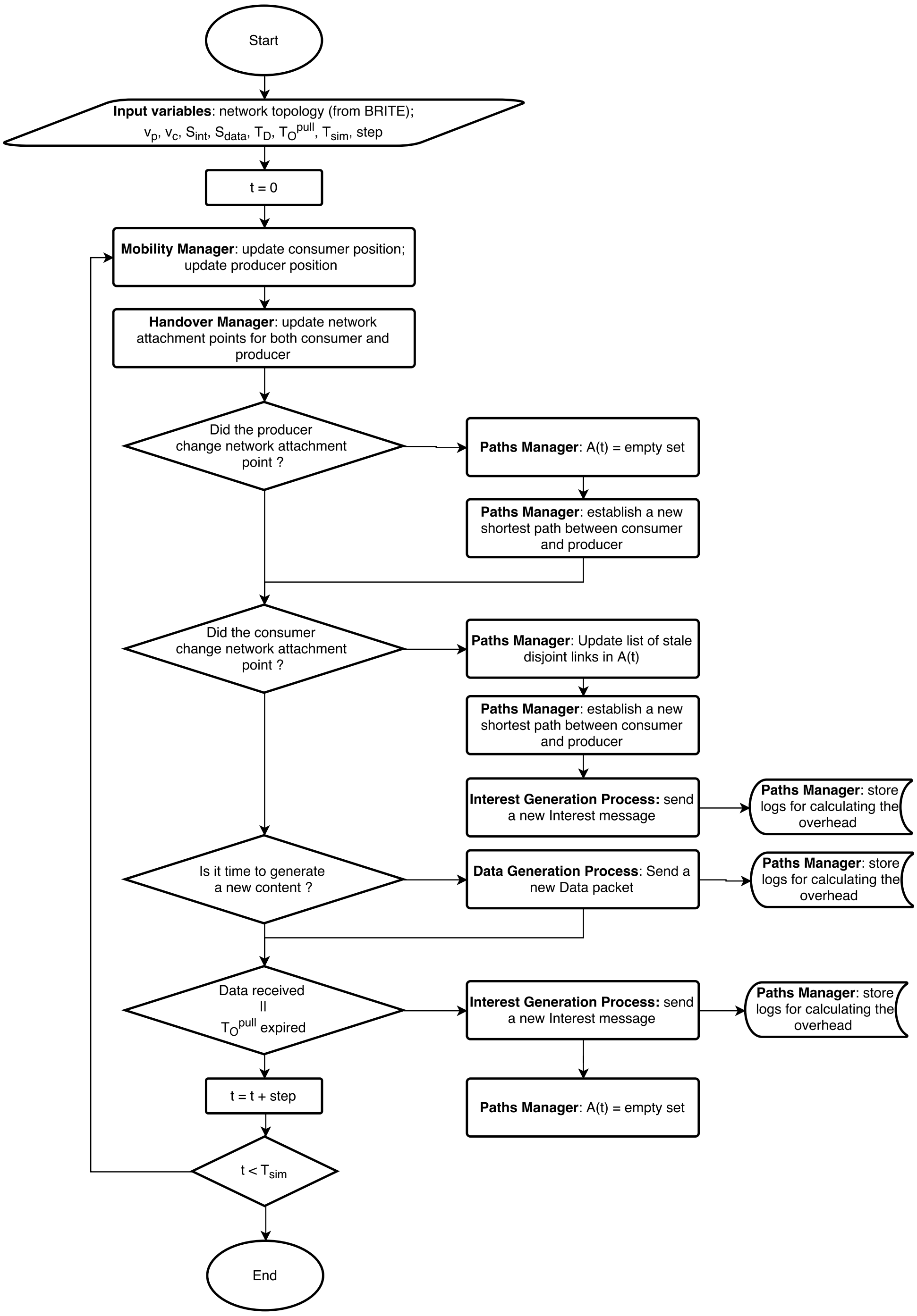


FIGURE 4.3: Algorithm used for implementing the pull-based approach in the ad-hoc simulator.

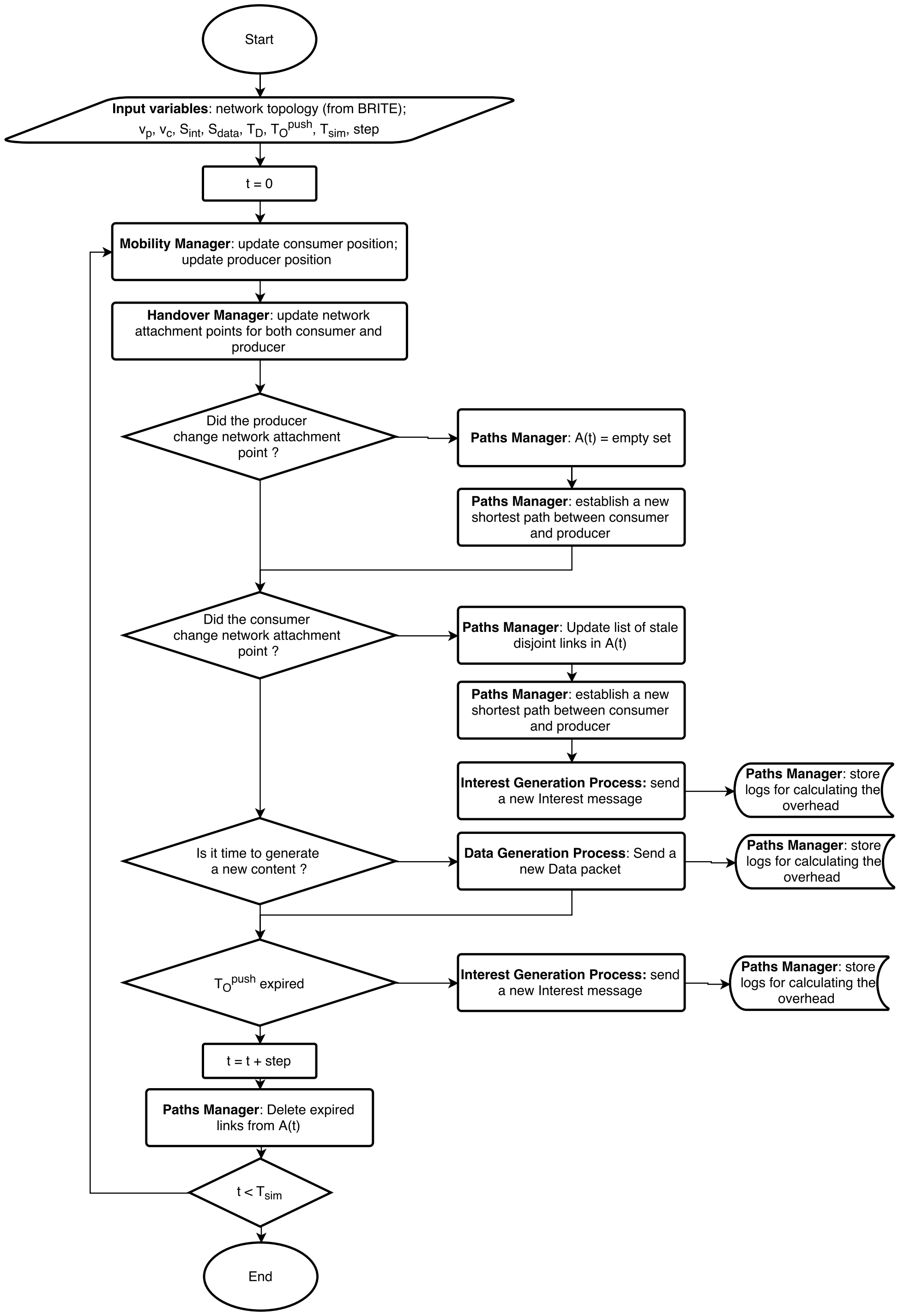


FIGURE 4.4: Algorithm used for implementing the push-based approach in the ad-hoc simulator.

4.3 Evaluation Results

Communication overhead generated by the pull-based approach, when the producer is stationary and the consumer is mobile.

Figure 4.5 shows the average communication overhead generated by the pull-based approach, obtained through analytical models and computer simulations in scenarios where the producer is stationary and the consumer is mobile.

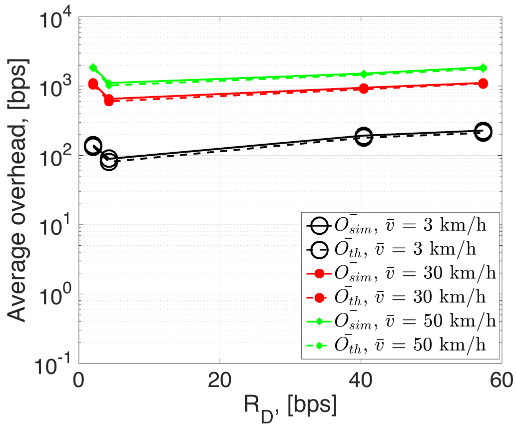
First of all, the higher the user speed, the higher the communication overhead. This result is due to the fact that when the consumer moves at a higher velocity, it may change the network attachment point more frequently. Indeed, the average cell residence time of the consumer, $\bar{\Delta t^C}$, reduces as well. Now, when $\bar{\Delta t^C}$ decreases, the number of *Interest* packets sent for establishing the multi-hop communication path between consumer and producer increases. At the same time, frequent handover processes provoke the generation of a higher number of stale paths. Therefore, the higher the user speed, the higher the contribution to the overhead due to extra data dissemination.

The content generation rate, and in more detail the average time interval between the generation of two consecutive contents (i.e., T_D) and *Data* packet size (i.e., S_D), also influences the communication overhead. IoT applications with lower T_D register a higher number of *Interest* packets transmitted by the mobile consumer. Moreover, the higher the application payload, the higher the contribution to the overhead due to the transmission of *Data* packet across stale disjoint links. In general, the communication overhead increases with the content generation rate. A counterintuitive result, instead, is given by the Smart Meter application. In this case, even if the application presents the lowest R_D, it does not reach the lowest communication overhead. From Table 4.2 it emerges that the Smart Meter application has a very high S_D value. Thus, the transmission of *Data* packet across stale disjoint links inflates the overall communication overhead.

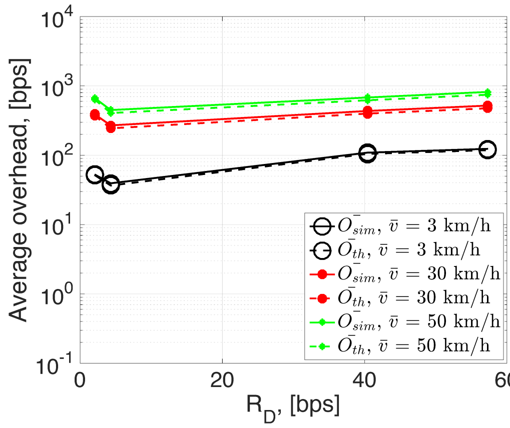
According to Eq. (3.4), the distance between consumer and producer strictly influences the communication overhead. Results demonstrate that the communication overhead increases with cell radius. In fact, the higher the cell radius, the lower the number of NDN nodes in the considered network topology. Therefore, while higher cell radius values bring to lower shortest path lengths, the communication overhead decreases as well. Urban and rural environments present different cell radius values. Specifically, urban environments have lower cell radius than urban ones. Thus, in line with the previous comments, urban scenarios always register higher communication overheads.

Communication overhead generated by the pull-based approach, when the producer is mobile and the consumer is stationary.

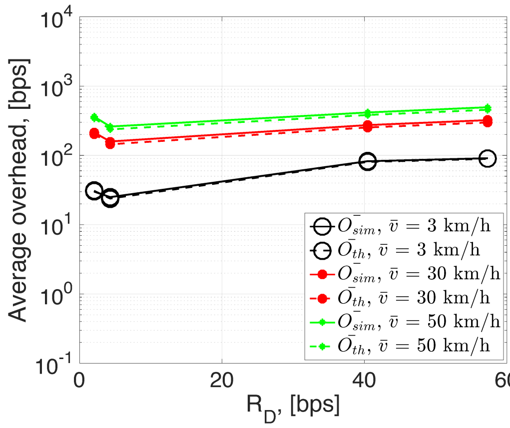
Figure 4.6 shows the average communication overhead generated by the pull-based approach, obtained through analytical models and computer simulations in scenarios where the producer is mobile and the consumer is stationary. As described in Eq. (3.9), the communication overhead is only



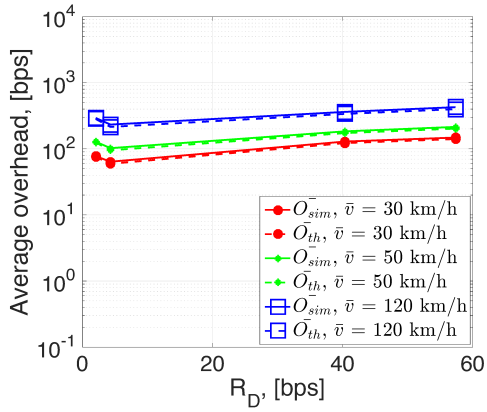
(a) Urban, r = 50 m



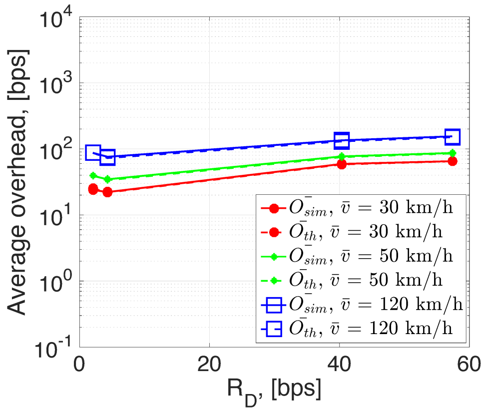
(b) Urban, r = 100 m



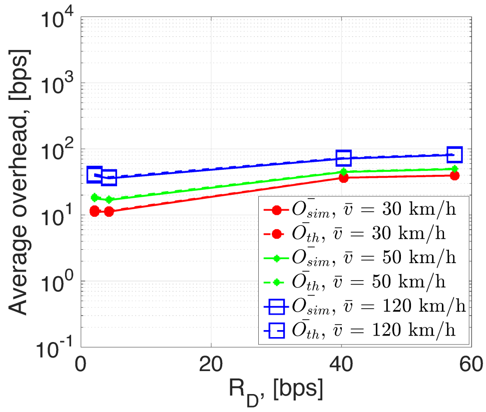
(c) Urban, r = 150 m



(d) Rural, r = 300 m

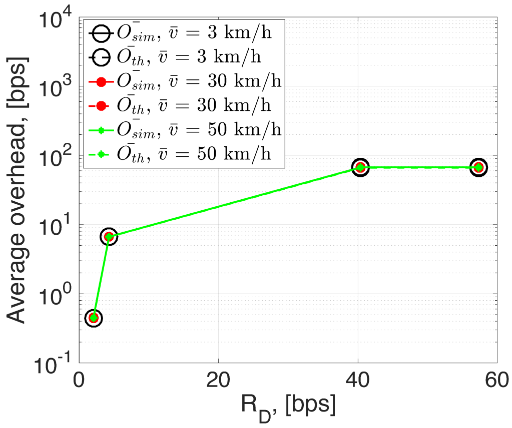


(e) Rural, r = 650 m

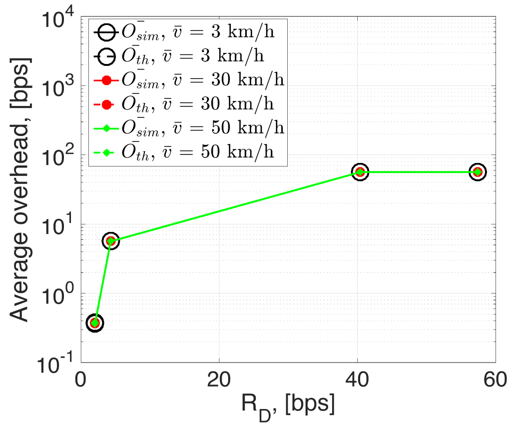


(f) Rural, r = 1000 m

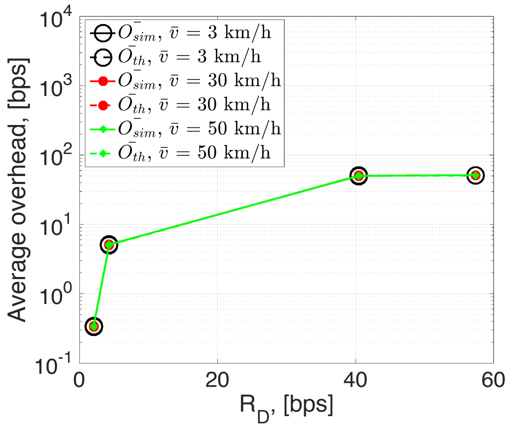
FIGURE 4.5: Communication overheads related to the pull-based approach, when the producer is stationary and the consumer is mobile.



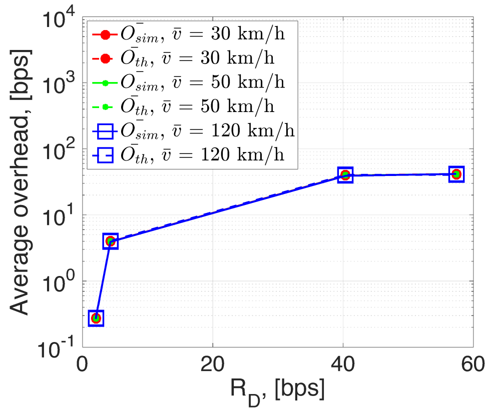
(a) Urban, r = 50 m



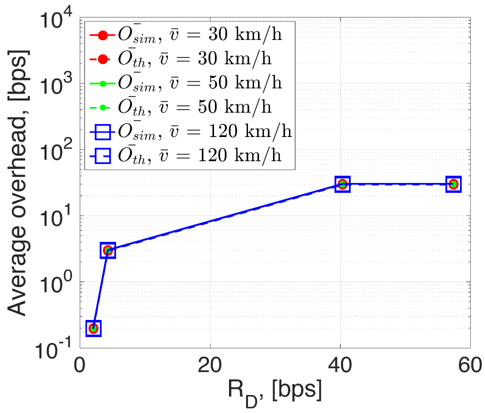
(b) Urban, r = 100 m



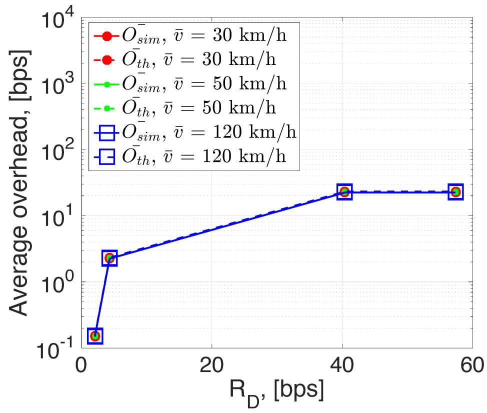
(c) Urban, r = 150 m



(d) Rural, r = 300 m



(e) Rural, r = 650 m



(f) Rural, r = 1000 m

FIGURE 4.6: Communication overheads related to the pull-based approach, when the producer is mobile and the consumer is stationary.

due to the transmission of control messages. *Interest* packets are sent every $T_O^{pull} = T_D$. Therefore, the resulting overhead increases as the average time interval between the generation of two consecutive contents (i.e., T_D) decreases. Since the lower T_D, the higher the average content generation rate (i.e., R_D), Figure 4.6 reports the average communication overheads that increases with R_D.

As already discussed, the distance between consumer and producer strictly influences the communication overhead: higher values of the cell radius imply lower numbers of NDN nodes and lower shortest path lengths. Thus, the average communication overhead decreases when the cell radius grows. Furthermore, as expected, urban scenarios always register higher communication overheads because of their lower cell radius values.

It is also important to highlight that the communication overhead does not depend on the producer speed. In fact, the producer speed does not influence the *Interest* generation process at the consumer side. Independently from the producer mobility, the consumer sends *Interest* packets when the timeout expires or when a new *Data* packet is retrieved. For this reason, the curves reported in Figure 4.6 overlap.

Communication overhead generated by the pull-based approach, when both producer and consumer are mobile.

Figure 4.7 shows the average communication overhead generated by the pull-based approach, obtained through analytical models and computer simulations in scenarios where both producer and consumer move at the same speed. Instead, Figure 4.8 reports the result related to the case where consumer and producer move at different speeds.

In general, all the considerations already argued for the previous two cases are still valid. First, the higher the consumer speed, the higher the communication overhead. When the consumer moves at a higher velocity, in fact, it may change the network attachment point more frequently and its average cell residence time, $\bar{\Delta t}^C$, reduces as well. This leads to a higher number of *Interest* packets sent for establishing the multi-hop communication path between consumer and producer. At the same time, frequent handover processes provoke the generation of a higher number of stale paths. Therefore, the higher the user speed, the higher the contribution to the overhead due to extra data dissemination. The content generation rate, and in more detail the average time interval between the generation of two consecutive contents (i.e., T_D) and *Data* packet size (i.e., S_D), also influences the communication overhead. IoT applications with lower T_D register a higher number of *Interest* packets transmitted by the mobile consumer. The distance between consumer and producer influences the communication overhead: higher values of the the cell radius imply lower numbers of NDN nodes and lower shortest path lengths.

In addition, by focusing the attention on Figure 4.8, it is possible to observe that lower values of the communication overhead are experienced when the producer moves faster than the consumer. As already anticipated in section 3, every time the producer changes its network attachment point, all the stale disjoint links do not produce overhead anymore. As a consequence (and differently from other scenarios) the average set of stale disjoint links, i.e., $\bar{\mathcal{A}}$, decreases and the average communication overhead reduces as well.

Communication overhead generated by the push-based approach, when the producer is stationary and the consumer is mobile.

Figure 4.9 shows the average communication overhead generated by the push-based approach, obtained through analytical models and computer simulations in scenarios where the producer is stationary and the consumer is mobile. From results it clearly emerges that the average communication overhead increases with the user speed. When the consumer moves at a higher velocity, it may change the network attachment point more frequently. Thus, the number of *Interest* packets sent for re-establishing a multi-hop communication path between consumer and producer increases as well. Moreover, the higher the user speed, the lower the average cell residence time of the consumer $\Delta\bar{t}^C$. In this context, the publish-subscribe approach is configured in order to ensure, with a probability close to 1, the presence of only one stale path every $\Delta\bar{t}^C$. Hence, while *Interest* life time decreases with $\Delta\bar{t}^C$, the corresponding generation rate increases with $\Delta\bar{t}^C$. As a result, the higher the user speed, the higher the impact of control messages to the communication overhead.

Of course, the transmission of *Data* packets across stale disjoint links inflates the communication overhead. But, its contribution is very low with respect to the one produced by control messages. For this reason, the communication overhead registers a slight increment with the average content generation rate.

As already discussed before, the communication overhead decreases with the cell radius and registers higher values in urban scenarios.

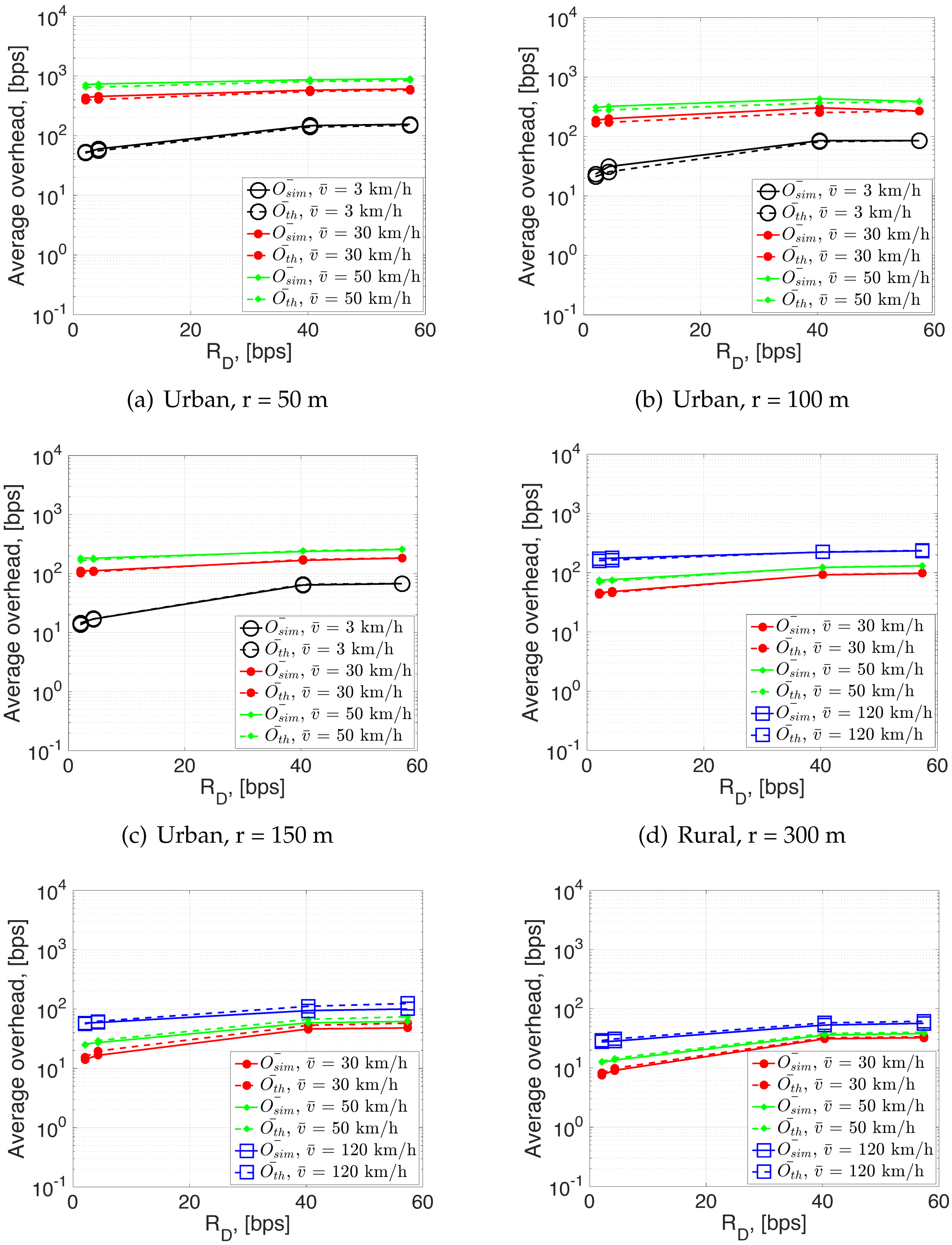
Communication overhead generated by the push-based approach, when the producer is mobile and the consumer is stationary.

Figure 4.10 shows the average communication overhead generated by the push-based approach, obtained through analytical models and computer simulations in scenarios where the producer is mobile and the consumer is stationary. As in the pull-based approach, also in this case the communication overhead is only due to subscription requests. However, when the push-based approach is used, the *Interest* generation process does not depend on the average content generation rate, but it is configured according to the user speed (see section 4.1.2). Indeed, the higher the user speed, the higher the *Interest* generation rate, the higher the resulting communication overhead.

Again, the communication overhead decreases with the cell radius and registers higher values in urban scenarios and reasons are the same of those widely explained in the previous paragraphs.

Communication overhead generated by the push-based approach, when both producer and consumer are mobile.

Figure 4.11 shows the average communication overhead generated by the push-based approach, obtained through analytical models and computer simulations in scenarios where both producer and consumer move at the same speed. Instead, Figure 4.13 reports the result related to the case where consumer and producer moves at different speed.



(a) Urban, r = 50 m

(b) Urban, r = 100 m

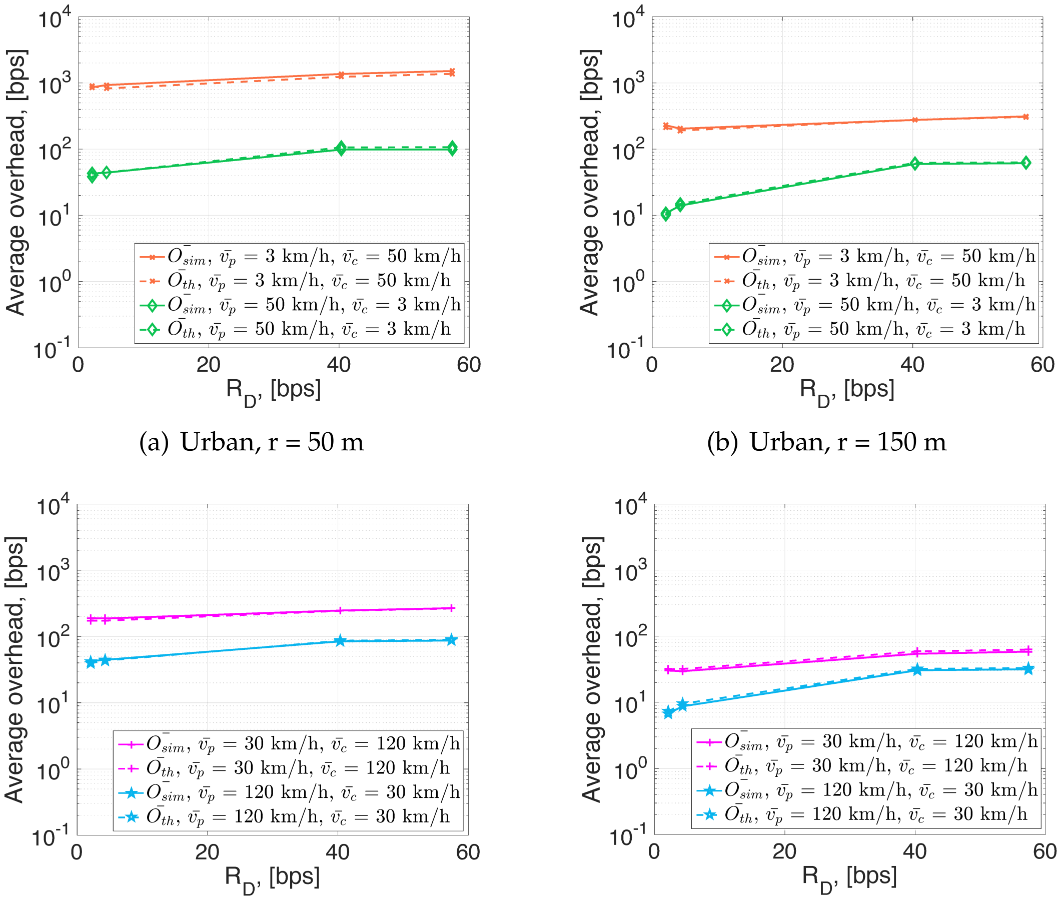
(c) Urban, r = 150 m

(d) Rural, r = 300 m

(e) Rural, r = 650 m

(f) Rural, r = 1000 m

FIGURE 4.7: Communication overheads related to the pull-based approach, when both producer and consumer are mobile and move with the same speed.



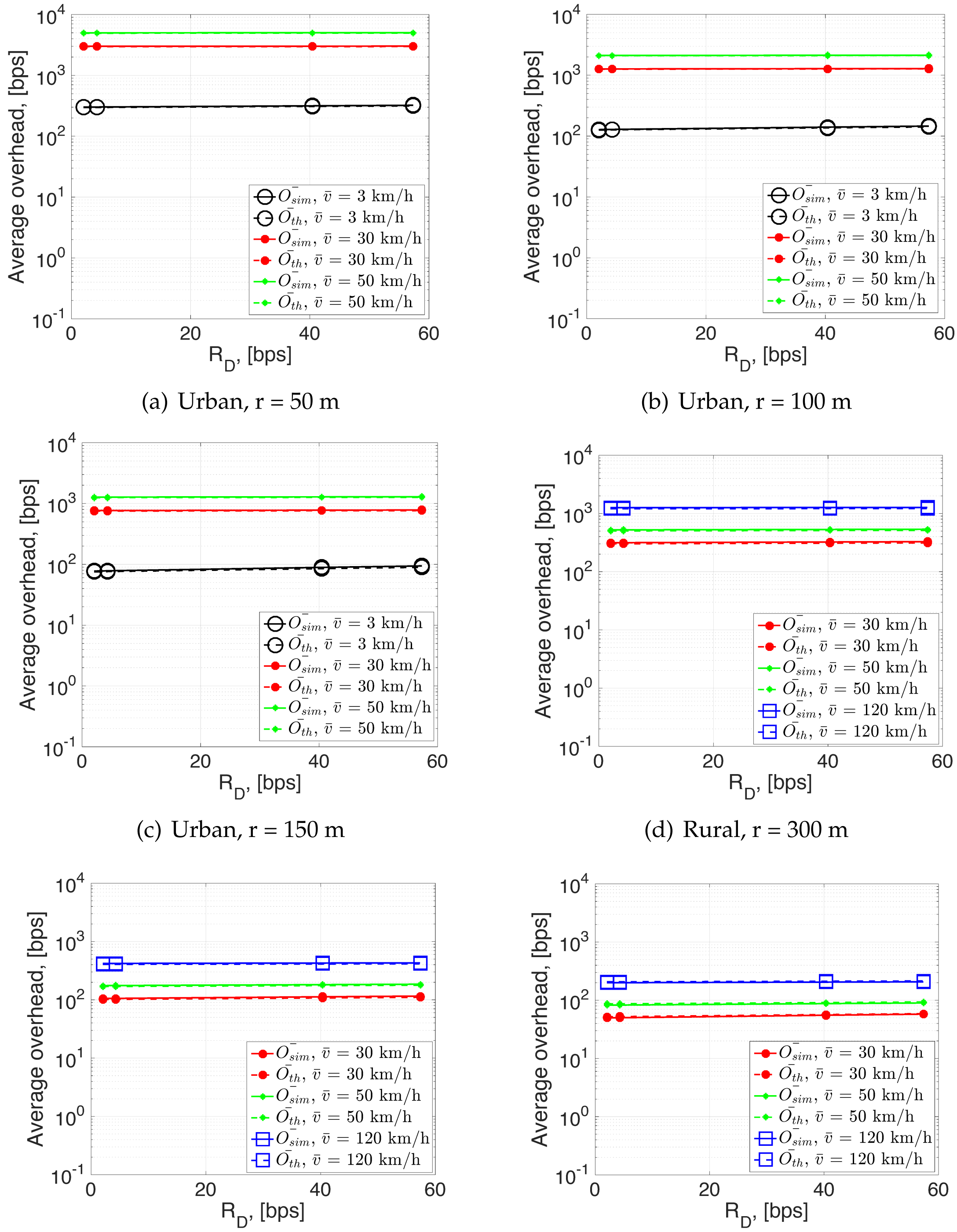
(a) Urban, r = 50 m

(b) Urban, r = 150 m

(c) Rural, r = 300 m

(d) Rural, r = 1000 m

FIGURE 4.8: Communication overheads related to the pull-based approach, when both producer and consumer are mobile and move with different speed.



(a) Urban, r = 50 m

(b) Urban, r = 100 m

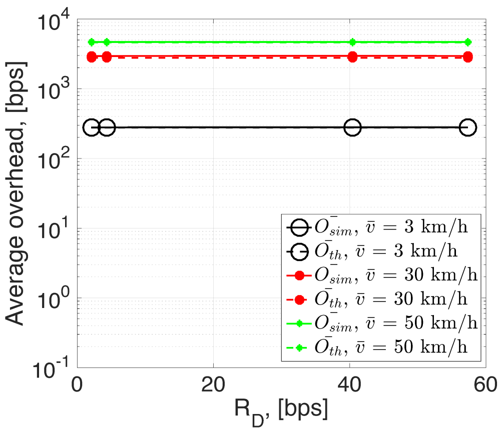
(c) Urban, r = 150 m

(d) Rural, r = 300 m

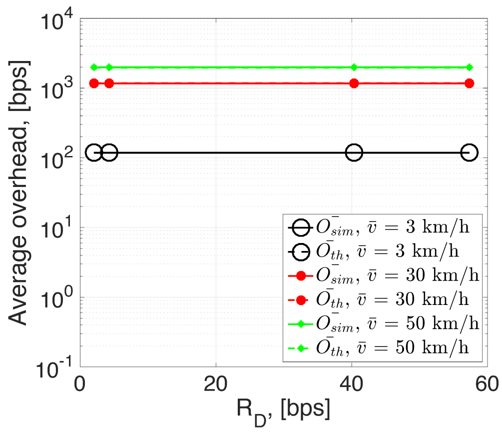
(e) Rural, r = 650 m

(f) Rural, r = 1000 m

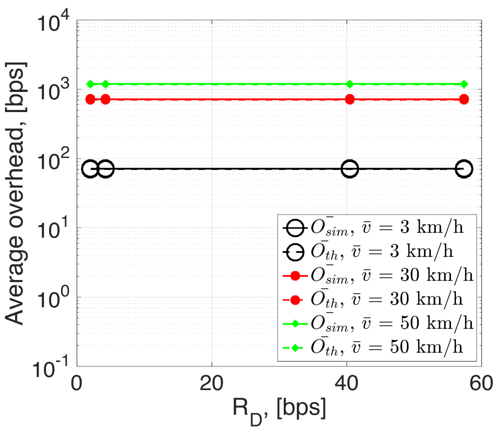
FIGURE 4.9: Communication overheads related to the push-based approach, when the producer is stationary and the consumer is mobile.



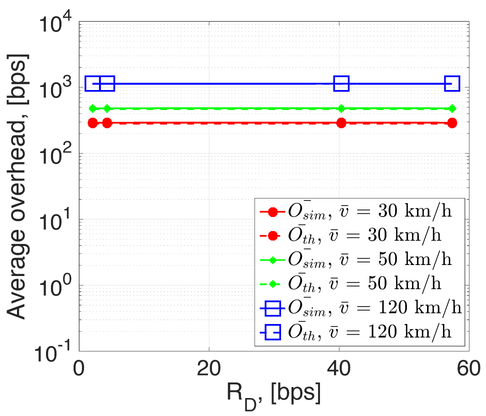
(a) Urban, r = 50 m



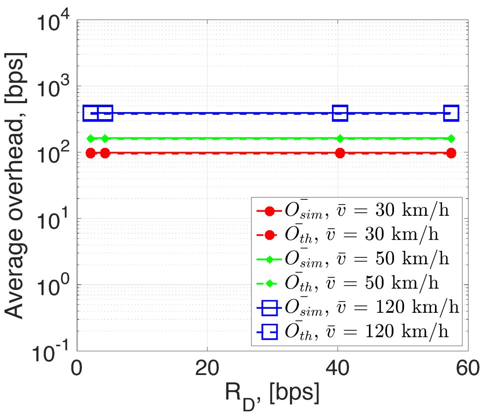
(b) Urban, r = 100 m



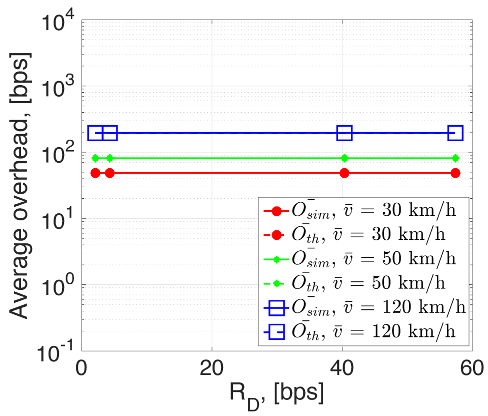
(c) Urban, r = 150 m



(d) Rural, r = 300 m



(e) Rural, r = 650 m



(f) Rural, r = 1000 m

FIGURE 4.10: Communication overheads related to the push-based approach, when the producer is mobile and the consumer is stationary.

As expected, the average communication overhead increases with the user speed. In fact, a higher velocity of the consumer leads to more frequent changes of network attachment point. Therefore, the number of *Interest* packets sent for re-establishing a multi-hop communication path between consumer and producer increases as well. Additionally, the speed of the consumer, the lower its average cell residence time $\Delta\bar{t}^C$. The publish-subscribe approach is configured in order to ensure, with a probability close to 1, the presence of only one stale path every $\Delta\bar{t}^C$. Hence, while the *Interest* life time decreases with $\Delta\bar{t}^C$, the corresponding generation rate increases with $\Delta\bar{t}^C$. As a result, the higher the user speed, the higher the impact of control messages to the communication overhead.

The contribution of the transmission of *Data* packets across stale disjoint links is lower than the one produced by control messages. Moreover, such a contribution decreases with the producer speed because of the reduction of the average set of stale disjoint links.

As already discussed before, the communication overhead decreases with the cell radius and registers higher values in urban scenarios.

Cross comparison

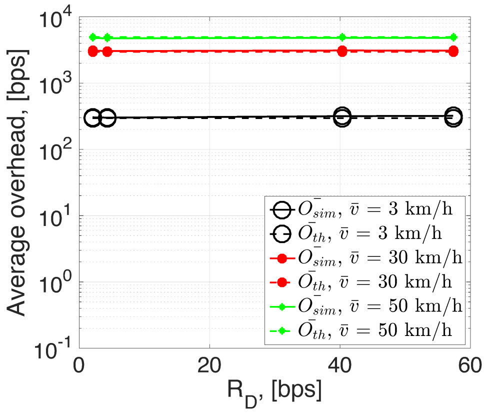
With reference to the set of results presented above, it is possible to observe that push-based and pull-based approaches share one common characteristic: the communication overhead they produce increases with the density of nodes belonging to the NDN network topologies.

When the consumer is mobile and the producer is stationary, both approaches register an increment of the average communication overhead with the user speed. Same considerations can be done for scenarios where both consumer and producer are mobile. In this case, however, the average communication overhead reduces because of the decrement of the average set of stale disjoint links. This behavior is mainly registered when the producer moves faster than the consumer. On the contrary, different behaviors are observed in a scenario with stationary consumer and mobile producer. Here, in fact, the user speed only impacts on the communication overhead related to the pull-based approach.

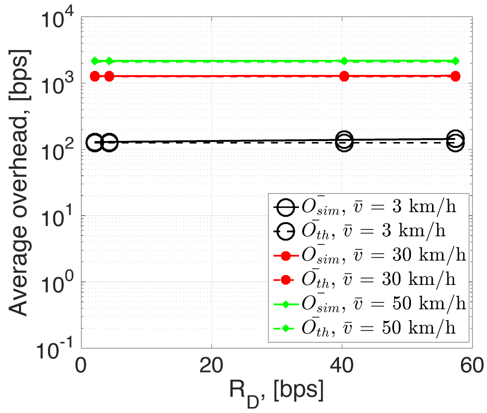
By observing the absolute values of both estimated and measured communication overheads, it is possible to conclude that the pull-based approach always guarantees the lowest bandwidth consumption, thus promising potential performance gains of the overall network performance. This result is due to the fact that the number of Interest messages sent is higher in the push-based approach than the pull-based one because of the timeouts set. Moreover, when the pull-based approach is used, stale paths remain active just for the transmission of a single Data packet and, as a consequence, the impact of extra data transmission, is significantly reduced.

4.3.1 Validation of analytical models

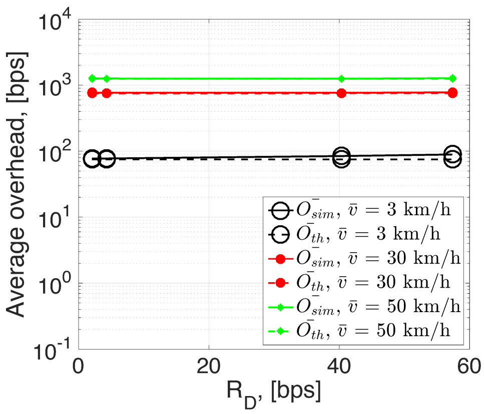
To provide a further insight, the accuracy of analytical models formulated in Chapter 3 is evaluated by means of the absolute relative error, calculated between the theoretical communication overhead and the one measured from



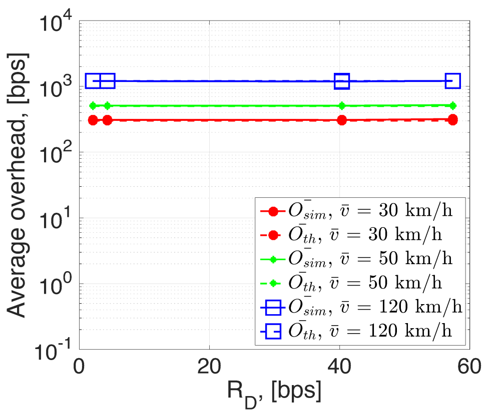
(a) Urban, r = 50 m



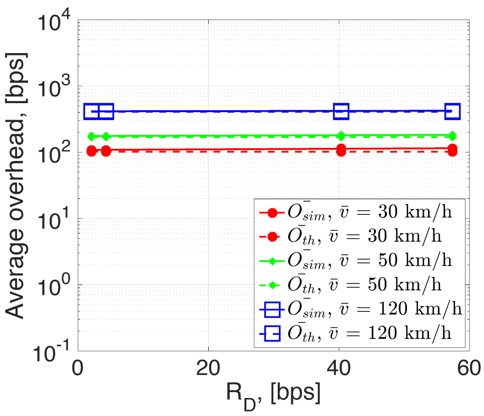
(b) Urban, r = 100 m



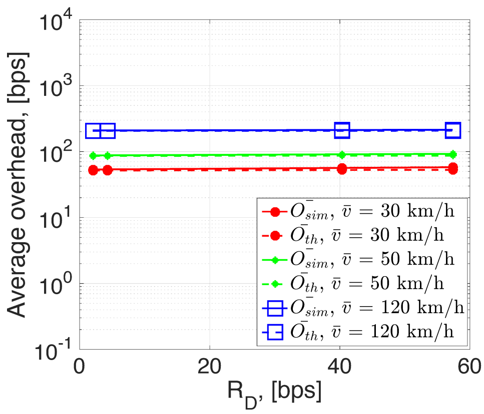
(c) Urban, r = 150 m



(d) Rural, r = 300 m

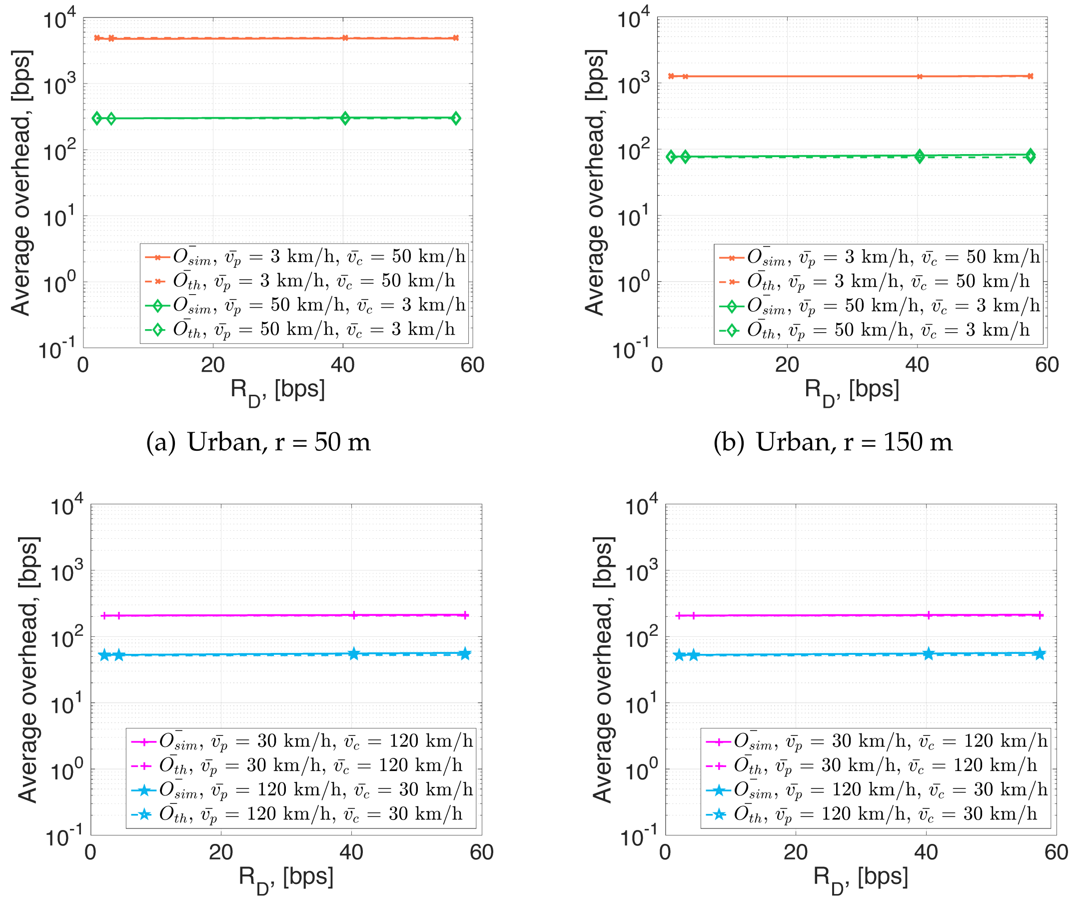


(e) Rural, r = 650 m



(f) Rural, r = 1000 m

FIGURE 4.11: Communication overheads related to the push-based approach, when both producer and consumer are mobile and move with the same speed.



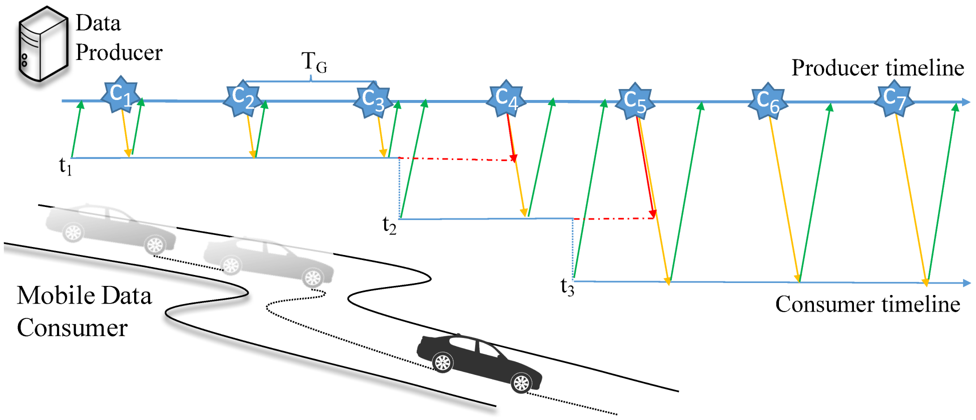
(a) Urban, r = 50 m

(b) Urban, r = 150 m

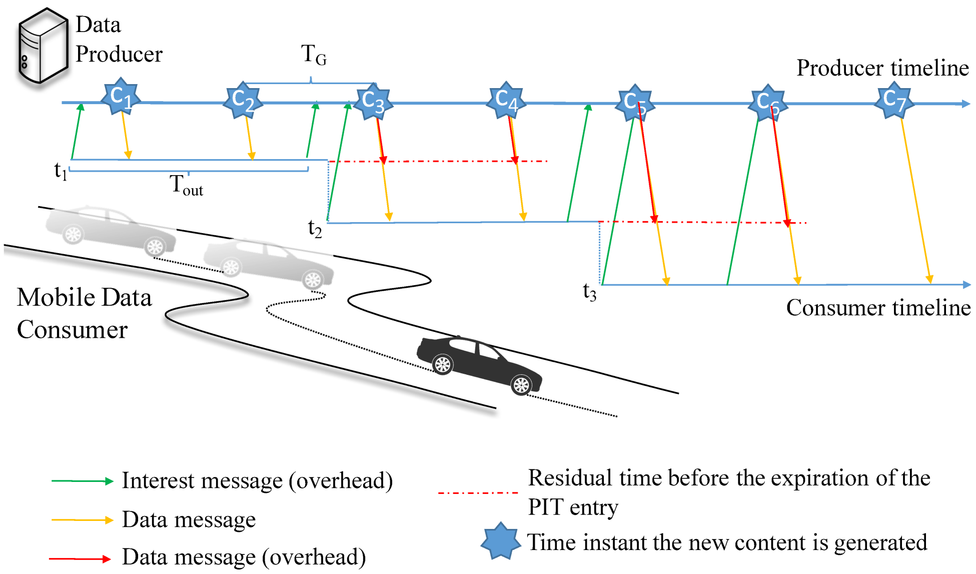
(c) Rural, r = 300 m

(d) Rural, r = 1000 m

FIGURE 4.12: Communication overheads related to the push-based approach, when both producer and consumer are mobile and move with different speed.



(a)



(b)

FIGURE 4.13: Message sequence chart showing *Interest* and *Data* packets exchanged when (a) the *pull-based* and (b) the *push-based* approach is used.

simulations, as defined in Eq. (4.2):

$$Error = \left| \frac{\bar{O_{sim}} - \bar{O_{th}}}{\bar{O_{sim}}} \right| 100, \tag{4.2}$$

where $\bar{O_{sim}}$ and $\bar{O_{th}}$ are the average communication overhead evaluated by simulations and the one estimated through analytical models, respectively.

The average and the maximum absolute relative errors related to results reported in sections 4.3-4.3, are shown in Table 4.3. Obtained values clearly demonstrate that the proposed analytical models are accurate enough to capture the behavior of the publish-subscribe mechanism based on the pull-based and push-based approach, in a wide range of scenarios.

TABLE 4.3: Average and maximum values for the absolute relative errors, calculated for both pull-based and push-based approaches.

	Stationary producer, mobile consumer	
	Pull-based	Push-based
Average error	4.80%	2.77%
Maximum error	9.53%	4.97%

	Mobile producer, stationary consumer	
	Pull-based	Push-based
Average error	2.07%	2.03 %
Maximum error	3.95%	6.53 %

	Mobile producer, mobile consumer	
	Pull-based	Push-based
Average error	4.75%	4.11%
Maximum error	10.13%	9.10 %

Conclusion

Starting from the investigation of Internet of Things applications' challenges and of the Information-Centric Networking's features, the ICN paradigm seems to have all the potential to enable IoT services. ICN can fulfill publish-subscribe data dissemination and natively supports mobile applications. But, ICN is not yet a completed architecture. In particular, the ICN-based solutions proposed in the literature to implement the publish-subscribe model do not explicitly address mobile scenarios.

Therefore, this thesis devised publish-subscribe communication mechanisms for Internet of Things applications in different mobile conditions. Built on top of the Information-Centric Networking paradigm, the developed solutions leverage both pull-based and push-based approaches. In addition, analytical models, describing the communication overhead that these models can introduce, were formulated and validated through computer simulations. To this end, benchmark network topologies, real Internet of Things applications, and standardized settings for urban and rural environments were taken into account.

Obtained results highlighted how the communication overhead is influenced by some of the parameters, which include cell radius, user speed, average content generation rate, and communication environment. Also, they allowed to determine pros and cons of pull-based and push-based approaches by underling the conditions under which one scheme should be preferred to the other one. In particular, the pull-based approach always guaranteed the lowest bandwidth consumption, thus promising potential performance gains of the overall network performance. This result is due to the fact that the number of Interest messages sent is higher in the push-based approach than the pull-based one because of the timeouts set. Moreover, when the pull-based approach is used, stale paths remain active just for the transmission of a single Data packet and, as a consequence, the impact of extra data transmission, is significantly reduced.

Finally, the absolute relative error, calculated between the theoretical communication overhead and the measured one, was always less than 10%. This clearly demonstrated the good level of accuracy offered by the proposed analytical models.

Future Work. This work is a merely starting point for understanding the possible benefits of the publish-subscribe communication mechanisms in ICN.

First of all, future research activities will further investigate the behavior of the conceived approaches considering scenarios with different kind of services, number of producers/consumers, mobility patterns, and routing algorithms.

Secondly, the implementation of new solutions can reduce the overhead due to mobility support. To this end, the Software Defined Networking (SDN) paradigm and the Network Function Virtualization (NFV) [98]-[101] can come to our aid. On one hand, NFV can provide the following advantages: (i) the separation of network functions from specific hardware (that allows the functions to run on a virtualized infrastructure) facilitates the deployment of new network protocols such as ICN; (ii) NFV orchestration eases the management of ICN services, because it ensures adequate compute, storage, and network resource to provide a specific network service. On the other hand, SDN, decoupling data forwarding plane and control plane, enables the entire network to be programmed, thus facilitating the configuration for ICN nodes, networks, applications, and services. Moreover, SDN provides a centralized tool that can ease the cache management and caching-based forwarding.

Finally, a comparison between ICN-based and IP-based publish-subscribe solutions can be investigated. Formulation, analysis, and performance evaluation of the communication overhead could help the scientific community understand if ICN can really outperform the TCP/IP architecture or identify the applications where it is more suitable.

Bibliography

[1] Cisco. *The Zettabyte Era: Trends and Analysis*. Tech. rep. Cisco Systems, Inc., 2017.

[2] Md Faizul Bari et al. "A survey of naming and routing in information-centric networks". In: *IEEE Communications Magazine* 50.12 (2012).

[3] Bengt Ahlgren et al. "A survey of information-centric networking". In: *IEEE Communications Magazine* 50.7 (2012).

[4] Giuseppe Piro et al. "Information-centric networking and multimedia services: present and future challenges". In: *Transactions on Emerging Telecommunications Technologies* 25.4 (2014), pp. 392–406.

[5] Kostas Pentikousis et al. "Information-centric networking [Guest editorial]". In: *IEEE Communications Magazine* 50.7 (2012), pp. 22–25.

[6] Jianli Pan, Subharthi Paul, and Raj Jain. "A survey of the research on future internet architectures". In: *IEEE Communications Magazine* 49.7 (2011).

[7] Teemu Koponen et al. "A Data-oriented (and Beyond) Network Architecture". In: *SIGCOMM Comput. Commun. Rev.* 37.4 (2007), pp. 181–192.

[8] Christian Dannewitz et al. "Network of information (netinf)–an information-centric networking architecture". In: *Computer Communications* 36.7 (2013), pp. 721–735.

[9] Nikos Fotiou et al. "Developing information networking further: From PSIRP to PURSUIT". In: *proc. of International Conference on Broadband Communications, Networks and Systems*. Springer. 2010, pp. 1–13.

[10] Ivan Seskar et al. "Mobilityfirst future internet architecture project". In: *Proceedings of the 7th Asian Internet Engineering Conference*. ACM. 2011, pp. 1–3.

[11] Gerardo García et al. "COMET: Content mediator architecture for content-aware networks". In: *IEEE Future Network & Mobile Summit (FutureNetw)*. 2011, pp. 1–8.

[12] Lixia Zhang et al. "Named data networking". In: *ACM SIGCOMM Computer Communication Review* 44.3 (2014), pp. 66–73.

[13] K. H. Yeh. "A Secure IoT-Based Healthcare System With Body Sensor Networks". In: *IEEE Access* 4 (2016), pp. 10288–10299.

[14] M. Zhu et al. "Public Vehicles for Future Urban Transportation". In: *IEEE Transactions on Intelligent Transportation Systems* 17.12 (2016), pp. 3344–3353.

[15] J. Lloret et al. "An Integrated IoT Architecture for Smart Metering". In: *IEEE Communications Magazine* 54.12 (2016), pp. 50–57.

[16] Giuseppe Piro et al. "Information Centric Services in Smart Cities". In: *Elsevier Journal of Systems and Software* 88 (2014), pp. 169–188.

[17] Yanyong Zhang et al. *Design Considerations for Applying ICN to IoT*. Internet-Draft. Internet Engineering Task Force, 2017. 40 pp.

[18] Wentao Shang et al. "Named data networking of things". In: *proc. of IEEE International Conference on Internet-of-Things Design and Implementation*. 2016, pp. 117–128.

[19] Marica Amadeo, Claudia Campolo, and Antonella Molinaro. "Information-centric networking for connected vehicles: a survey and future perspectives". In: *IEEE Communications Magazine* 54.2 (2016), pp. 98–104.

[20] Nikos Fotiou and George C Polyzos. "Realizing the Internet of Things using information-centric networking". In: *proc. of IEEE 10th International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (QShine)*. 2014, pp. 193–194.

[21] Luigi Alfredo Grieco et al. "Architecting information centric ETSI-M2M systems". In: *proc. of IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*. 2014, pp. 211–214.

[22] J. Chen et al. "Exploiting ICN for Realizing Service-Oriented Communication in IoT". In: *IEEE Communications Magazine* 54.12 (2016), pp. 24–30.

[23] Sugang Li et al. "IoT Middleware Architecture over Information-Centric Network". In: *IEEE Globecom Workshops (GC Wkshps)*. 2015, pp. 1–7.

[24] Patrick Th Eugster et al. "The many faces of publish/subscribe". In: *ACM computing surveys (CSUR)* 35.2 (2003), pp. 114–131.

[25] Dominique Guinard et al. "From the internet of things to the web of things: Resource-oriented architecture and best practices". In: *Architecting the Internet of things*. Springer, 2011, pp. 97–129.

[26] Akram Hakiri et al. "Publish/subscribe-enabled software defined networking for efficient and scalable IoT communications". In: *IEEE Communications Magazine* 53.9 (2015), pp. 48–54.

[27] Luigi Atzori, Antonio Iera, and Giacomo Morabito. "The internet of things: A survey". In: *Computer networks* 54.15 (2010), pp. 2787–2805.

[28] Michele Zorzi et al. "From today's intranet of things to a future internet of things: a wireless-and mobility-related view". In: *IEEE Wireless Communications* 17.6 (2010).

[29] Giuseppe Piro et al. "Gazing into the crystal ball: when the Future Internet meets the Mobile Clouds". In: *IEEE Transactions on Cloud Computing* PP.99 (2016), pp. 1–1.

[30] Andrea Detti et al. "Exploiting content centric networking to develop topic-based, publish-subscribe MANET systems". In: *Ad Hoc Networks* 24 (2015), pp. 115–133.

[31] Christos Tsilopoulos and George Xylomenos. "Supporting diverse traffic types in information centric networks". In: *proc. of ACM SIGCOMM workshop on Information-centric networking*. 2011, pp. 13–18.

[32] Jiachen Chen et al. "Copss: An efficient content oriented publish/subscribe system". In: *proc. of ACM/IEEE Symposium on Architectures for Networking and Communications Systems*. 2011, pp. 99–110.

[33] Jérôme François, Thibault Cholez, and Thomas Engel. "CCN traffic optimization for IoT". In: *IEEE Fourth International Conference on the Network of the Future (NOF)*. 2013, pp. 1–5.

[34] Marica Amadeo, Claudia Campolo, and Antonella Molinaro. "Internet of things via named data networking: The support of push traffic". In: *proc. of IEEE International Conference and Workshop on the Network of the Future*. 2014, pp. 1–5.

[35] Alberto Medina et al. "BRITE: An approach to universal topology generation". In: *proc. of IEEE Ninth International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*. 2001, pp. 346–353.

[36] Luís ML Oliveira and Joel JPC Rodrigues. "Wireless Sensor Networks: A Survey on Environmental Monitoring". In: *JCM* 6.2 (2011), pp. 143–151.

[37] Huawei. *Global Connectivity Index 2016*. Tech. rep. Huawei Technologies Co., Ltd., 2016.

[38] TELECOMMUNICATION STANDARDIZATION SECTOR OF ITU. *Recommendation ITU-T Y.2060*. Tech. rep. 2012.

[39] G. Piro et al. "Gazing into the crystal ball: when the Future Internet meets the Mobile Clouds". In: *IEEE Transactions on Cloud Computing* (2016), pp. 1–1. DOI: `10.1109/TCC.2016.2573307`.

[40] Guillermo Barrenetxea et al. "Wireless sensor networks for environmental monitoring: the sensorscope experience". In: *proc. of IEEE International Zurich Seminar on Communications*. 2008, pp. 98–101.

[41] Bratislav Predić et al. "Exposuresense: Integrating daily activities with air quality using mobile participatory sensing". In: *proc. of IEEE International Conference on Pervasive Computing and Communications (PERCOM) Workshops*. 2013, pp. 303–305.

[42] Zhengguo Sheng et al. "A survey on the ietf protocol suite for the internet of things: Standards, challenges, and opportunities". In: *IEEE Wireless Communications* 20.6 (2013), pp. 91–98.

[43] George Xylomenos et al. "A survey of information-centric networking research". In: *Communications Surveys & Tutorials, IEEE* 16.2 (2014), pp. 1024–1049.

[44] William Tärneberg, Vishal Chandrasekaran, and Marty Humphrey. "Experiences creating a framework for smart traffic control using AWS IOT". In: *Proceedings of the 9th International Conference on Utility and Cloud Computing*. ACM. 2016, pp. 63–69.

[45] Jere Backman et al. "IoT-based interoperability framework for asset and fleet management". In: *IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*. 2016, pp. 1–4.

[46] Muhammad Alam, Joaquim Ferreira, and José Fonseca. "Introduction to intelligent transportation systems". In: *Intelligent Transportation Systems*. Springer. 2016, pp. 1–17.

[47] Colrain M Zuppo. "Defining ICT in a boundaryless world: The development of a working hierarchy". In: *International journal of managing information technology* 4.3 (2012), p. 13.

[48] Roger R Stough. *Intelligent Transport Systems*. Edward Elgar Publishing, 2001.

[49] Bob McQueen and Judy McQueen. *Intelligent transportation systems architectures*. 1999.

[50] Fatih Sakiz and Sevil Sen. "A survey of attacks and detection mechanisms on intelligent transportation systems: VANETs and IoV". In: *Ad Hoc Networks* 61 (2017), pp. 33–50.

[51] B Ayyappan and P Mohan Kumar. "Vehicular Ad Hoc Networks (VANET): Architectures, methodologies and design issues". In: *Science Technology Engineering and Management (ICONSTEM), Second International Conference on*. IEEE. 2016, pp. 177–180.

[52] ISO. *Intelligent transport systems, Cooperative ITS, Using V2I and I2V communications for applications related to signalized intersections.* Tech. rep. ISO/AWI TS 19091:2017. 2017.

[53] ETSI. *Technical Specification Intelligent Transport Systems (ITS)*. Tech. rep. 102 637-1 V1. 1.1. 2010.

[54] Marica Amadeo, Claudia Campolo, and Antonella Molinaro. "Enhancing content-centric networking for vehicular environments". In: *Computer Networks* 57.16 (2013), pp. 3222–3234.

[55] M Aguilera Leal et al. "Information-centric opportunistic data dissemination in vehicular ad hoc networks". In: *International IEEE Conference on Intelligent Transportation Systems (ITSC)*. 2010, pp. 19–22.

[56] John Day. "How in the heck do you lose a layer!?" In: *Network of the Future (NOF), 2011 International Conference on the*. IEEE. 2011, pp. 135–143.

[57] John Day. *Patterns in network architecture: a return to fundamentals*. Pearson Education, 2007.

[58] Jerome Saltzer. *On the naming and binding of network destinations*. Tech. rep. 1993.

[59] Al-Mukaddim Khan Pathan and Rajkumar Buyya. "A taxonomy and survey of content delivery networks". In: *Grid Computing and Distributed Systems Laboratory, University of Melbourne, Technical Report* 4 (2007).

[60] Prakash Suthar, Anil Jangam, and Dirk Trossen. *Native Deployment of ICN in LTE, 4G Mobile Networks*. Internet-Draft. Internet Engineering Task Force, 2017. 40 pp.

[61] Sugang Li et al. "A comparative study of MobilityFirst and NDN based ICN-IoT architectures". In: *Heterogeneous Networking for Quality, Reliability, Security and Robustness (QShine), 2014 10th International Conference on*. IEEE. 2014, pp. 158–163.

[62] Mohamed Ahmed M Hail et al. "On the performance of caching and forwarding in information-centric networking for the IoT". In: *International Conference on Wired/Wireless Internet Communication*. Springer. 2015, pp. 313–326.

[63] Boubakr Nour et al. "A Distributed ICN-Based IoT Network Architecture: An Ambient Assisted Living Application Case Study". In: *GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE. 2017, pp. 1–6.

[64] M Ascigil et al. "A keyword-based ICN-IoT platform". In: *Proceedings of the 4th ACM Conference on Information-Centric Networking*. ACM. 2017, pp. 22–28.

[65] Z Zhu et al. "Recent advances in connected vehicles via information-centric networking". In: (2016).

[66] Kamrul Hasan and Seong-Ho Jeong. "CCN-based vehicular communications". In: *Information Networking (ICOIN), 2018 International Conference on*. IEEE. 2018, pp. 926–929.

[67] George Xylomenos et al. "Caching and mobility support in a publish-subscribe internet architecture". In: *IEEE Communications Magazine* 50.7 (2012).

[68] Marica Amadeo et al. "Information-centric networking for the internet of things: challenges and opportunities". In: *IEEE Network* 30.2 (2016), pp. 92–100.

[69] Yu Zhang et al. "A survey of mobility support in named data networking". In: *proc. of IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. 2016, pp. 83–88.

[70] Agnese V Ventrella, Giuseppe Piro, and L Alfredo Grieco. "Publish-subscribe in mobile information centric networks: Modeling and performance evaluation". In: *Computer Networks* 127 (2017), pp. 317–339.

[71] Chao Fang et al. "A survey of mobile information-centric networking: Research issues and challenges". In: *IEEE Communications Surveys & Tutorials* (2018).

[72] Yongqiang Huang and Hector Garcia-Molina. "Publish/subscribe in a mobile environment". In: *Wireless Networks* 10.6 (2004), pp. 643–652.

[73] Agnese V Ventrella, Giuseppe Piro, and Luigi Alfredo Grieco. "Information-centric publish-subscribe mechanisms for Intelligent Transportation Systems". In: *Service Operations and Logistics, and Informatics (SOLI), 2017 IEEE International Conference on*. IEEE. 2017, pp. 199–204.

[74] A. Rahman et al. *Deployment Considerations for Information-Centric Networking (ICN)*. Internet-Draft. Internet Engineering Task Force, 2018. 40 pp.

[75] Xuan Nam Nguyen, Damien Saucez, and Thierry Turletti. "Providing CCN functionalities over OpenFlow switches". In: (2013).

[76] Dirk Trossen et al. "Ip over icn-the better ip?" In: *Networks and Communications (EuCNC), 2015 European Conference on*. IEEE. 2015, pp. 413–417.

[77] G Carofiglio. *Mobile video delivery with Hybrid ICN*. Tech. rep. Cisco, Tech. Rep, 2016.

[78] Agnese V Ventrella, Luigi Alfredo Grieco, and Giuseppe Piro. "Information-Centric Networking in Environmental Monitoring: an overview on publish-subscribe implementations". In: *Advanced Video and Signal Based Surveillance (AVSS), 2017 14th IEEE International Conference on*. IEEE. 2017, pp. 1–6.

[79] Ali Shariat, Ali Tizghadam, and Alberto Leon-Garcia. "An ICN-based publish-subscribe platform to deliver UAV service in smart cities". In: *proc. of IEEE Conference on Computer Communications Workshops*. 2016, pp. 698–703.

[80] Antonio Carzaniga, Michele Papalini, and Alexander L Wolf. "Content-based publish/subscribe networking and information-centric networking". In: *Proceedings of the ACM SIGCOMM workshop on Information-centric networking*. 2011, pp. 56–61.

[81] Tomohiko Yagyu et al. "Demo: Content-based push/pull message dissemination for disaster message board". In: *ACM Proceedings of the 2nd International Conference on Information-Centric Networking*. 2015, pp. 205–206.

[82] Wassim Drira and Fethi Filali. "A Pub/Sub extension to NDN for efficient data collection and dissemination in V2X networks". In: *proc. of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*. 2014, pp. 1–7.

[83] David Tipper. "Resilient network design: challenges and future directions". In: *Telecommunication Systems* 56.1 (2014), pp. 5–16.

[84] Konstantinos Katsaros et al. "Information-centric networking for machine-to-machine data delivery: a case study in smart grid applications". In: *IEEE Network* 28.3 (2014), pp. 58–64.

[85] Matheus B Lehmann, Marinho P Barcellos, and Andreas Mauthe. "Providing producer mobility support in NDN through proactive data replication". In: *proc. of IEEE/IFIP Network Operations and Management Symposium (NOMS)*. 2016, pp. 383–391.

[86] Bohao Feng, Huachun Zhou, and Qi Xu. "Mobility support in Named Data Networking: a survey". In: *EURASIP Journal on Wireless Communications and Networking* 2016.1 (2016), p. 220.

[87] Zhenkai Zhu, Alexander Afanasyev, and Lixia Zhang. "A new perspective on mobility support". In: *Named-Data Networking Project, Tech. Rep* (2013).

[88] Yan Zhang. "Handoff performance in wireless mobile networks with unreliable fading channel". In: *IEEE Transactions on Mobile Computing* 9.2 (2010), pp. 188–200.

[89] Kumudu S Munasinghe and Abbas Jamalipour. "Analysis of signaling cost for a roaming user in a heterogeneous mobile data network". In: *proc. of IEEE Global Telecommunications Conference (GLOBECOM)*. 2008, pp. 1–5.

[90] FANTASTIC5G. *Deliverable D2.1 Air interface framework and specification of system level simulations*. Tech. rep. D2.1. Call: H2020-ICT-2014-2, Project reference: 671660, Flexible Air iNTerfAce for Scalable service delivery wiThin wIreless Communication networks of the 5th Generation (FANTASTIC-5G), 2016.

[91] Kwan L Yeung and Sanjiv Nanda. "Channel management in microcell/macrocell cellular radio systems". In: *IEEE Transactions on Vehicular Technology* 45.4 (1996), pp. 601–612.

[92] Quentin De Coninck et al. "Observing real smartphone applications over multipath TCP". In: *IEEE Communications Magazine* 54.3 (2016), pp. 88–93.

[93] Rapeepat Ratasuk, Jun Tan, and Amitava Ghosh. "Coverage and capacity analysis for machine type communications in LTE". In: *IEEE Vehicular Technology Conference (VTC Spring)*. 2012, pp. 1–5.

[94] Yi Wang et al. "Fast name lookup for named data networking". In: *proc. of IEEE 22nd International Symposium of Quality of Service (IWQoS)*. 2014, pp. 198–207.

[95] Alexander Afanasyev et al. "SNAMP: Secure namespace mapping to scale NDN forwarding". In: *proc. of IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. 2015, pp. 281–286.

[96] Van Jacobson et al. *Named data networking (NDN) project 2013-2014 report*. 2014.

[97] Albert-László Barabási. *Network science*. Cambridge University Press, 2016.

[98] Markus Vahlenkamp et al. "Enabling ICN in IP networks using SDN". In: *Network Protocols (ICNP), 2013 21st IEEE International Conference on*. IEEE. 2013, pp. 1–2.

[99] N Blefari Melazzi et al. “An openflow-based testbed for information centric networking”. In: *2012 Future Network & Mobile Summit (FutureNetw)*. IEEE. 2012, pp. 1–9.

[100] Ravishankar Ravindran et al. “Towards software defined icn based edge-cloud services”. In: *Cloud Networking (CloudNet), 2013 IEEE 2nd International Conference on*. IEEE. 2013, pp. 227–235.

[101] Thanh Dinh and Younghan Kim. *Considerations for Using SDN/NFV in ICN*. Internet-Draft. Internet Engineering Task Force, 2018.