Universidade de Aveiro
2021

**Diego
Caldeira Hernandez**

Conhecimento da mobilidade do consumidor em redes centradas em informação

Consumer mobility awareness in information centric networks

Universidade de Aveiro
2021

Diego
Caldeira Hernandez

Conhecimento da mobilidade do consumidor em redes centradas em informação

Consumer mobility awareness in information centric networks

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de Computadores e Telemática, realizada sob a orientação científica do Doutor Nuno Miguel Abreu Luís, Professor Adjunto do Instituto Superior de Engenharia de Lisboa, e da Doutora Susana Isabel Barreto de Miranda Sargento, Professora Catedrática do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro.

o júri / the jury

presidente / president — Professor Doutor Amaro Fernandes de Sousa
Professor Auxiliar do Departamento de Eletónica, Telecomunicações e Informática da Universidade de Aveiro (por delegação do Vice-Reitor da Universidade de Aveiro)

vogais / examiners committee — Professor Doutor Bruno Miguel de Oliveira Sousa
Professor Auxiliar Convidado da Universidade de Coimbra (arguente)

Professora Doutora Susana Isabel Barreto de Miranda Sargento
Professora Catedrática do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro (co-orientadora)

agradecimentos / acknowledgements

Aos meus pais, quero deixar o meu profundo e maior agradecimento pelo seu imensurável carinho, apoio e tremendo sacrifício pessoal que fizeram a fim de proporcionar aos seus filhos a melhor educação e futuro possível. Faltarão sempre palavras para descrever a minha gratidão.

Um enorme obrigado ao professor Doutor Miguel Luís e à professora Doutora Susana Sargento por toda a orientação, apoio e disponibilidade dispensada para o desenvolvimento desta dissertação, como também pela oportunidade que me deram nestes últimos dois anos ao me acolherem no grupo de Investigação *Network Architecture and Protocols* (NAP), lugar onde consegui aprender e desenvolver-me nas mais variadas áreas técnicas e científicas.

Posteriormente, quero também agradecer ao Doutor Carlos Senna e ao Mestre Luís Gameiro pelos vários comentários que teceram e enriqueceram o trabalho desenvolvido.

Um obrigado a todos os meus amigos, que acompanharam de perto o meu percurso académico, parte importante pela qual o mesmo foi tão especial.

Por fim, agradeço ao Instituto de Telecomunicações e ao Fundo Europeu de Desenvolvimento Regional (FEDER), através do Programa Operacional Competitividade e Internacionalização (COMPETE 2020) do Portugal 2020 e do Programa Operacional Regional de Lisboa, e à Fundação para a Ciência e Tecnologia no âmbito do projetos MobiWise (POCI-01-0145-FEDER-016426) e InfoCent-IoT (POCI-01-0145-FEDER-030433).

Palavras Chave

Internet do Futuro, Mobilidade, Rede Centradas em Informação, *Named Data Networking*, Estimação de Mobilidade, *Single-Data request*, *Publish-subscribe*

Resumo

O tráfego de dados móveis tem vindo a crescer significativamente, sobretudo devido à evolução das tecnologias de comunicação sem fios, o que tem vindo a implicar o desenho e implementação de novos e diferentes tipos de redes móveis.
Os paradigmas de redes centradas em informação têm sido apontados como uma alternativa para contornar as restrições impostas pelas redes tradicionais IP, nomeadamente a mobilidade dos seus utilizadores. Apesar das potenciais vantagens em relação aos ambientes móveis sem fios, vários desafios de investigação ainda necessitam de ser resolvidos, mais especificamente aqueles relacionados com o processo de handover dos seus utilizadores móveis, levando por vezes à perda de informação.
Esta dissertação tem como objetivo o desenvolvimento de mecanismos de suporte à mobilidade do Consumidor para redes ICN, utilizando duas abordagens distintas de comunicação: solicitação única de conteúdo e o modelo $publish - subscribe$. Os esquemas propostos exploram uma entidade remota de previsão de mobilidade, cujo objetivo é monitorizar e antecipar eventuais trajetórias de posição dos utilizadores móveis, obrigando a infraestrutura a ajustar-se aos novos caminhos do consumidor, resultando numa forma eficiente de gestão de mobilidade dos utilizadores com o objetivo de garantir uma melhor qualidade de serviço.
A implementação e avaliação dos esquemas propostos foi realizada utilizando o $ndnSIM$, em cenários funcionais e não funcionais. Estes últimos utilizam registos reais de mobilidade e conetividade urbana. Os resultados obtidos mostram que a solução proposta ultrapassa significativamenta a versão nativa do NDN e as soluções tradicionais de $publish - subscribe$, considerando a taxa de entrega de conteúdos e sobrecarga da rede.

Keywords

Future Internet, Mobility, Information-Centric Network, Named Data Networking, Mobility Predictor, Single-data request, Publish-subscribe.

Abstract

Mobile data traffic is expanding significantly since the surge and evolution of wireless communication technologies, leading to the design and implementation of different types of mobile networks.

Information Centric Network paradigms have been pointed as an alternative to bypass the restrictions imposed by the traditional IP Networks, such as the one imposed by the mobility of its users. Despite their potential advantages regarding mobile wireless environments, several significant research challenges remain to be addressed, more specifically the communication damage due to handover, causing loss of packets.

The scope of this dissertation is the development of NDN-based mechanisms with support for Consumer mobility in two different communication approaches: single content request and publish-subscribe. The proposed schemes address a remote mobility predictor entity, whose purpose is to monitor and anticipate trajectories, while compelling the infrastructure to adjust to the new paths, resulting in an efficient way to manage the consumers' mobility with the purpose of attaining a better quality of service to users.

The implementation and evaluation of the proposed schemes were performed using $ndnSIM$, through functional and non-functional scenarios. The latter uses real traces of urban mobility and connectivity. The obtained results show that the proposed solution far surpasses the native NDN workflow and the traditional publish-subscribe solutions with respect to content delivery ratio and network overhead.

Contents

List of Figures

List of Tables

Glossary

ICN Information Centric Networks
RSU Road Side Unit
OBU On Board Unit
NDN Named Data Network
CS Content Store
PIT Pending Information Table
FIB Forwarding Information Base
NACK Negative Acknowledgement
TLV Type Length Value
CCN Content Centric Networks
TCP Transmission Control Protocol
UDP User Datagram Protocol
IP Internet Protocol
NDNLP Named Data Network Link Protocol
NLSR Named Data Link State Routing Protocol
RSSI Received signal strength indication
MAC Media Access Control
LSA Link State Advertisement
RIB Routing Information Base
MTU Maximum Transmission Unit
iHEMS Information-Centric Approach to Secure Home Energy Management
NS-3 Network Simulator
NFD Named Data Networking Forwarding Daemon
NDNSim Named Data Network Simulator
CSMA Carrier-Sense Multiple Access
MM Mobility Manager
SIS Subscription Interest Switch
SDS Subscription Data Switch
AP Acess Point
IS Interest Switch
DS Data Switch
ISP Internet Service Provider
P2P Peer-to-peer
CDN Content Delivery Network
ISP Internet Service Provider
CP Content Provider
DONA Data Oriented Network
PURSUIT Publish-Subscribe Internet Routing Paradigm
URL Uniform Resource Locator
URI Uniform Resource Identifier
NAR Network Access Router
VANET Vehicular Adhoc Networks
MANET Mobile Ad-Hoc Networks
V2V Vehicle-to-Vehicle Communication
CP Content Provider
MLS Mobile Link Service
MNC Mobile Node Consumer
V2I Vehicle-to-Infrastructure
V2V Vehicle-to-Vehicle
BSC Base Station Controller
BTS Base Transceiver Stations
IoT Internet of Things
OTT Over the Top
RTT Round Trip Time
WAVE Wireless Access in Vehicular Environments

CHAPTER 1

Introduction

This chapter reports the context and motivation for the work covered in this dissertation. It describes the main objectives and contributions, as well as the document structure.

1.1 Context & Motivation

The technology evolution is allowing the increase of portable communication devices, and consequently, their dominance over the Internet traffic. Their mobility raises challenges for the underlying network to support them. These challenges and demands are expected to increase with the advent of Internet of Things (IoT) and 5G, where billions of IoT devices, many of them mobile, are expected to be connected in the near future to achieve many future applications, such as smart cities [1].

From a network infrastructure point of view, mobility involves the physical and topological re-location of a device. Location-oriented networks are frequently dependant on connection-oriented protocols, for example, the Transmission Control Protocol (TCP). This means that, in order to settle communication between two entities, the establishment of a session between client and server needs to be paired. The reason for this is mainly due to the non-capability of both parties to understand what should be sent and received at any given time. Therefore, a session is established, and it is required for both parties to know each others' network addresses to assure communication reliability. When client mobility is introduced, re-establishment of these sessions is required, so that both parties are aware of the up-to-date network addresses, causing severe disruption and degradation of the quality of the service provided to the user. A network is pronounced to have support of Consumer mobility if it grants the users re-configuration of their network location without disrupting connectivity.

The Internet has been evolving very rapidly and used primarily for content distribution. Research works in the literature have proposed the re-design of the current Internet's architecture to be Information-Centric [2], replacing host-to-host routing infrastructure with a content-based one. The routing is based on the use of unique content identifiers to optimal sources, making content a first-class entity. Hence, Information Centric Networks (ICN) does not necessarily suffer from the aforementioned constraints. Communications are explicitly made at the network level: when a Consumer generates a request for a specific content, it knows exactly what it should receive in return, without necessarily requiring the Content Provider (CP) cooperation, thus the establishment of persistent sessions is unnecessary, and the re-location of a host does not demand for the re-establishment of a connection.

Among the different ICN architectures, NDN is the one indicated as the promising future Internet architecture, able to support content location independent communication and content dispersion with use of in network caching [3]. In this architecture Consumer mobility is provided by the re-transmission of the content request at the new location. However, there are drawbacks such as the re-transmission of content request and redundant transmissions of content.

Some of the research regarding smart cities makes use of intelligent entities, working as mobility forecasters providing maintenance of client's connections, and thus handover management, proven to be an efficient resource capable of enhancing the quality of service to the mobile Consumers. Therefore, this work is focused on building schemes to achieve Consumer mobility in a NDN-based architecture, by making use of such intelligent entities.

1.2 Objectives

The objectives for this dissertation are the study, implementation and testing of seamless content delivery to mobile Consumers through ICN-based communication architectures. The proposed schemes adapt and extend concepts from information-centric networks, by means of disseminating and caching the content at new recipients in order to serve the Consumer already established. This requires the inquiry for the new location, which will be calculated by a remote mobility predictor entity. These solutions cover, not only the single inquiry of content, but also the publish-subscribe communication type. This strives for a better quality of content delivery to Consumers. As such, the objectives are as follows:

- Study on literature on mechanisms handling seamless handover or retrieval of lost content due to the damaged communication path caused by the mobility of the Consumer, with a particular focus on the ones implemented for NDN;
- Design and implementation of different strategies to address seamless retrieval of content, tackling the communication damage induced by the handover of the mobile Consumer using single request of content and publish-subscribe communication approaches;
- Evaluation of proposed strategies, assessing their efficiency through functional scenarios, with synthetic mobility, and scenarios with real mobility traces of vehicles (using a real mobile dataset from Oporto city).

1.3 Contributions

The work addressed in this dissertation led to the following contributions:

- Study of the implications from having mobile clients in communication networks, from both Internet Protocol (IP) and ICN environments;

- Study of the NDN architecture and its suitability for environments with mobile Consumers;
- Design and implementation of a centralized network entity that is aware of information about mobile Consumers and their mobility through time, with the aim to use it for modelling the mobility prediction;
- Design of different communication approaches to disseminate mobility predictions to the entities responsible to assure the communication towards the new location of the mobile Consumer, considering single request of content and publish-subscribe communication approaches;
- Design and implementation of support packets with the aim to establish content deviation or bifurcation towards the new location of the Consumer, in order to achieve seamless handover;
- Design and implementation of a mechanism with support for Consumer mobility in an NDN based architecture, without the needs of using IP, by approaching a remote mobility predictor entity to establish seamless handover. It also covers seamless publish-subscribe communication for mobile content subscribers;
- Fabrication of mobility traces for functional scenarios, as well as the analysis, pre-processing and filtering of real mobility traces in order to achieve veracity in the obtained results when simulating associated scenarios;
- Simulation testing scenario of mobile Consumers in a NDN-based network with seamless handovers in functional scenarios and scenarios presenting real mobility and connectivity patterns.

Part of the work developed in this dissertation, namely the one addressing the publish-subscribe Consumer mobility, was submitted to the IEEE Internet of Things Journal with the title *Handling Producer and Consumer Mobility in IoT Publish-Subscribe Named Data Networks*, and is currently under revision (Major revisions). A second paper on the *Single-Data-Request* with mobility awareness approach is being prepared.

1.4 Document structure

The remaining document is structured as follows:

- **Chapter 2 - State of the Art**: this chapter presents a general analysis and description of the fundamental concepts related to this dissertation: Consumer mobility and its impact in current network infrastructures; Information-Centric Networks and their relevant architectures. The related work addressing recovery of damage communication due to Consumer handover in an ICN environment is also presented;
- **Chapter 3 - Proposed Solutions**: this chapter describes the proposed schemes to address seamless handover for Consumers in a NDN-based architecture, including the description of the entities, their mechanisms and interactions;
- **Chapter 4 - Implementation**: this chapter describes the implementation of the proposed modifications in the NDN architecture, and consequently in the simulation platform, including the technical aspects of integration;

- **Chapter 5 - Evaluation**: it presents the evaluation performed on the implemented solution through the simulation platform results, in which some of the presented scenarios include real-world vehicular mobility traces;
- **Chapter 6 - Conclusion**: it presents the conclusions of the dissertation, the next steps to be taken to improve the proposed schemes, and future guidelines.

CHAPTER 2

State of the Art

The increase of mobile communication devices is evolving significantly. The surge of wireless technologies made it possible for the design and implementation of different types of networks, such as Vehicular Adhoc Networks (VANET). The mobility of end-users can damage the communication sessions when no additional protocol is established. Therefore, there is the need to overcome these difficulties to ensure a specific quality of service to the clients.
The work in this dissertation aims at the design, implementation and evaluation of strategies that support seamless content retrieval for mobile Consumers in an ICN.
In this chapter, the ICN is introduced as a new paradigm for the future internet. A brief presentation of its common key features is made. Furthermore, the chapter proceeds by shortly describing the most relevant architectures in the current research, one of them being the base network model employed for this work: the Named Data Network (NDN). Then, the NDN limitations and challenges imposed by the mobility of the end-user Consumers are addressed at the design level. Finally, this chapter finalizes with related works approaching strategies with the purpose of handling interrupted communication due to handover situations.

2.1 Mobility

With the significant growth in wireless gadgets such as phones, tablets, and others, the mobile data traffic has expanded significantly [4]. Besides that, people are more interested on the desired content, rather than the location it comes from.

The above mentioned gadgets are equipped with wireless communication network technologies. These technologies allow them to communicate with each other and the Internet while being unbound to a location. Hence, topologies are constantly changing because of the intermittent mobile devices' connectivity, which are caused by entering and exiting networks.

Since the introduction of these wireless technologies, several mobile architectures have appeared, some of them are:

- *Cellular Networks*: mobile nodes connect to multiple Acess Point (AP) or Base Stations according to the positions they take. They are a long-range, high speed communication technology, whose coverage is granted by these fixed Base Stations.

- *Mobile Ad-Hoc Networks*: mobile nodes are network nodes with shared routing information, allowing message exchange between each other, as well as external parties via APs [5].
- *Vehicular Ad-Hoc Networks*: employ the principles of Mobile Ad-Hoc Networks (MANET) to the domain of vehicles.

The work developed under the scope of this dissertation makes use of VANETs as a use case scenario, from implementation to evaluation chapters. Hence, it is the responsibility of this chapter to present to the reader a more detailed description about this architecture and the technologies used.

2.1.1 Vehicular Ad-Hoc Networks

VANETs [6] consist of vehicles equipped with On Board Unit (OBU)s, which are intelligent units responsible for the dynamic dissemination of data with other vehicles and the network infrastructure, by resorting to their wireless interfaces.

The OBUs communicate with the infrastructure through RSUs. RSUs have wireless communication interfaces, such as Wi-Fi and Wireless Access in Vehicular Environments (WAVE), and are connected to the wired infrastructure through cable or fiber. Hence, they work as a bridge for vehicles to communicate with an Internet Service Provider (ISP) and vice-versa. Furthermore, an OBU hosted in a vehicle allows its passengers' and others nearby communication devices, paired with these units, to have Internet connectivity. Therefore, RSUs and OBUs can operate as a provider and as an user of services [7].

A variety of vehicles can be deployed for this type of networks, as observed in Figure 2.1, for example: cars, ambulances, buses, trucks, bikes and drones.

The commonly used type of communications are the following [8]:

- **Pure Ad-Hoc**: represents Vehicle-to-Vehicle (V2V). A node reaches other nodes using neighbouring vehicles without the support of the wired infrastructure. This communication is represented by the purple arrows in Figure 2.1.
- **Vehicle-to-Infrastructure (V2I)**: OBUs communicate through RSUs to access other networks and remote nodes. Such communication is represented by the pink arrows in Figure 2.1.
- **Mobile Infrastructure**: OBUs can serve as Mobile Gateways in order to allow distant nodes to access the networks.

VANETs may come as a solution to tackle mobility concerns. However, challenges such as intermittent connectivity regarding V2I, still need to be addressed when the focus is efficient and reliable delivery of content to Consumers. The work that this document covers solely focuses on the issues concerning V2I communications.

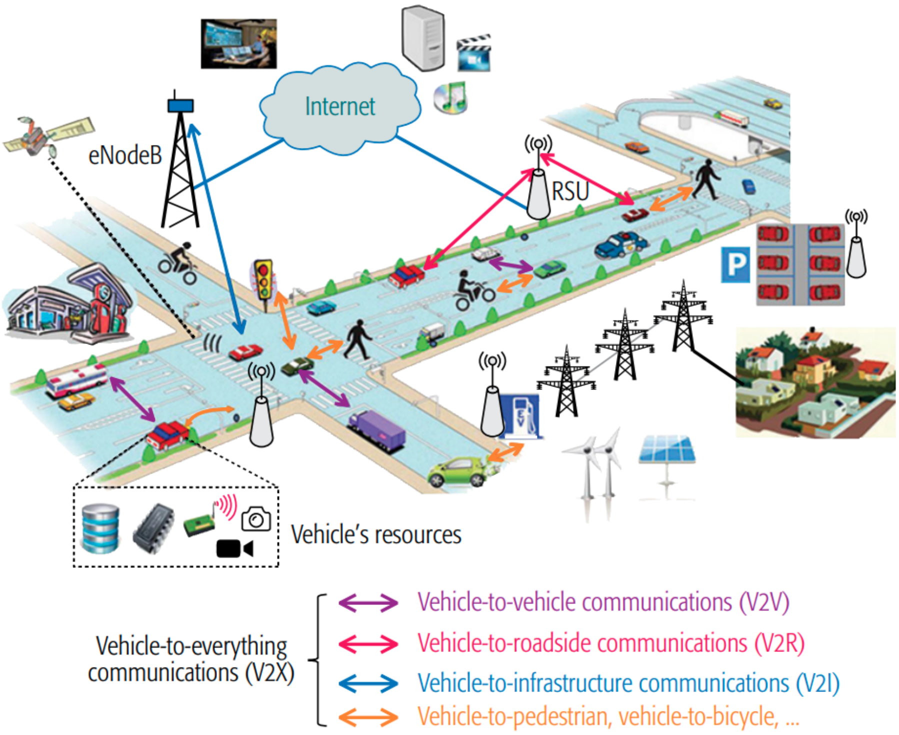


Figure 2.1: Vehicular communications in a Smart City context [6].

2.1.2 Consumer handover

Handovers or handoffs are known as the process of transferring a mobile node's communication session from a network to another, due to the transition to a new base station or access point. This requires the network configurations and network resources to reroute communication to the new base station reflecting the device's new location [9]. For instance, there is the need for the involvement of special protocols whose purpose is to guarantee seamless handover, and thus, a suitable quality of service. To achieve efficiency, it must detect the instant for the new communication settlement and forwarding mechanism to handle the request and content retrieval during this process. Ideally, mobility should not result in loss data or extended periods of disconnection.

Figure 2.2 shows an example of hard handover [10]. When the mobile device is between both base stations (during hard handover phase, in Figure 2.2), it switches with any of the base stations. The source cell is released, and only then the channel in the target cell is engaged (after hard handover phase, in Figure 2.2). A Base Station Controller (BSC) is a mediator responsible for the allocation of radio channels, receiving measurements from mobile devices, controlling Base Transceiver Stations (BTS) to BTS handovers and setup calls.

Location-oriented networks like IP did not anticipate the aforementioned mobility issues. Moreover, current IP network stack does not have an explicit understanding of the data involved. Hence, bilateral agreement is needed to ensure the retrieval of the correct data. To carry on with the share of information between both client and server, both parties need to be aware of their current network addresses. Therefore, a reestablishment of connections, like

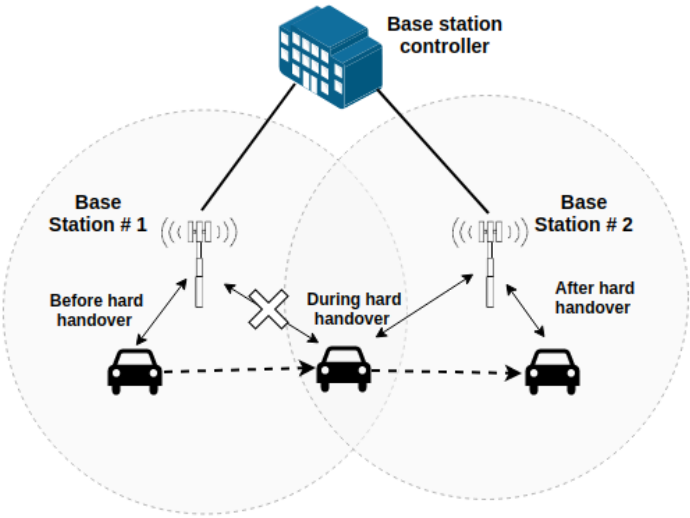


Figure 2.2: Example of a hard handover.

TCP, due to the mobility of the user, would cause overhead and damage to the communication session.

Mobile IP [11] and *Host Identification Protocol* [12] are some of the developed mechanisms which attempt to maintain consistency in the node's network address, causing an undesirable overhead [13].

A shift towards ICN-based paradigms is getting attention by the research community to address these issues. Its main concept does not force applications to take on host-centric information, allowing transmission of content without including network-layer address. Moreover, the communication model becomes receiver-driven, with no need to establish a communication session with the CP to achieve reliability from the interested content.

However, some of the core problems relating to mobility are not solved but rather shifted, remaining a number of important issues in this domain to be solved.

2.2 Information Centric Network (ICN)

The Internet has become an important infrastructure technology in today's society. It is evolving rapidly, with the popularization of optical fiber applications, improved network bandwidth, and the explosive increase rate of network traffic due to the popularity of many applications, such as streaming services.

Many technologies have been introduced to efficiently manage large volumes of information, improve spectrum efficiency, and increase network capacity. Besides these technologies, the traditional TCP/IP still has major problems in the face of increasing traffic [14]: routing process, scalability and content sharing performance.

Concepts such as Peer-to-peer (P2P) [15] and Content Delivery Network (CDN) [16] improve content sharing and distribution in the Internet [17]. P2P is an application-specific protocol, where content is delivered from end-users, whereas CDN can be a proprietary solution working at the application layer, operating at a proprietary infrastructure.

A blueprint of ICN [18] as the Internet system structure is a needed revolution that could completely change the current Internet in terms of scalability, security and mobility, therefore satisfying the needs of future social development.

In contrast to P2P and CDN, ICN is a standardized protocol working at the network layer, and the content is delivered from the infrastructure itself. It improves network performance and facilitates the retrieval and replication of content using in-network content caching.

The following sections will cover an overview of fundamental concepts and features of ICN. Moreover, they will address the most remarkable ICN proposed architectures, including the one used as a base work on this dissertation: Named Data Networking (NDN).

2.2.1 Overview

ICN introduces new communication paradigms for efficient content delivery. It follows the next two principals: the content is located using a name rather than an IP of the host; every ICN node should be capable of caching and serve the requested content.

The communication follows a receiver-driven approach, in which the content traverses the reverse route of the request. Therefore, it is the responsibility of the system to track the requested data location and send it back to the Consumer. Different content has its unique location-independent name. The fact that the content is searched by its designation grants ICN the deployment of in-network caching and content replication to improve the delivery of information.

Network caching can provide benefits from the point of view of the Consumer, CP and ISP [19]. Some of them are:

- Caching content in ISP cache fastens the content delivery. Then, the quality of service for the end-users is enhanced;
- Caching eases up the outgoing traffic of packets to other ISPs or CPs, diminishing the backhaul bandwidth consumption. Specially, it lowers the load on the CP when the request of contents are satisfied from the ISP cache preventing network congestion;
- When the CP is temporarily offline, the ISP cache is able to provide the content to end-users. Thus, it improves the robustness of the network.

2.2.2 ICN key features

Many research works have focused on different ICN functionalities such as naming, in-network caching and mobility. This section describes briefly the key features of ICN.

Content naming

A crucial key feature of ICN is the content Name. It identifies the content itself and enables the means to be searched for. It can be compact, persistent, scalable, and is able to validate the content [20]. The naming scheme allows name aggregation. Four relevant types of naming schemes have been introduced in ICN:

- **Hierarchical Names** [21] consist of multiple components, similar to the current user-friendly structure of the Uniform Resource Identifier (URI). They identify the content and describe the application/services and they allow aggregation of similar names, boosting the scalability potential of the architecture by reducing the routing table size. An example of the naming aggregation is shown in Figure 2.3;

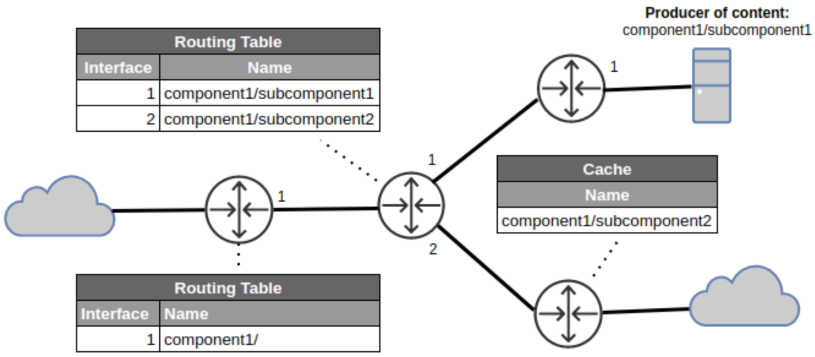


Figure 2.3: Example of hierarchical naming aggregation.

- **Flat Names** [22] consist of fixed, globally unique and self-certified names. Usually, they are obtained through hash-based algorithms applied to the content, being non-human readable. They allow the use of more efficient lookup algorithms, such as Bloom Filters. However, the aggregation is difficult, since it makes routing tables more complex;
- **Attribute-value based Names** [23] have a collection of attributes, where each one posses a name, a type, and a set of possible values. This scheme promotes an easier search for content, by moving from full-on name lookup to inquiry of keywords that are associated with the content;
- **Hybrid Names** [21] combine at least two of the previously denoted schemes. They use the best features that allow the improvement in the network, such as scalability, performance, security and privacy enhancement.

The naming scheme will have a direct impact on the routing process, from operations such as name lookup, forwarding decisions and management of routing tables. These implications may be more severe for constrained devices, such as the ones from IoT environments.

Routing and forwarding

The content is independent from its original location, and the content name is the basis to discover and forward the data back to the Consumer. Its design is receiver-driven: hop-by-hop, the request is forwarded until it reaches the original or replica node which has the requested data. The content is then composed as a packet and delivered to the inquirer, following the reverse path.

An efficient ICN routing and forwarding process strives to have the following properties [24]:

- **Content state**: routing should provide low-latency network operations for content, whether it is the original or a replica. These operations include registration, meta-data update, lookup and deletion of the content. These operations maintain the integrity of the content, such as freshness and availability, within the network;

- **Discovery of closest copy**: to diminish inter-domain traffic, the routing mechanism should always consider the shortest route towards the content (original or a replica), according to the established network metrics;
- **Discovery guarantee**: locating any content should be guaranteed regardless its popularity and replication degree;
- **Scalability**: routing mechanisms will have to support great magnitudes of content names, which are to accommodate future growth. The routing table size and routing path length should be as short as possible, considering the number of different contents and the memory limitations of technologies.

In-network caching

During communications, an ICN node is able to cache content with the aim of serving it for future requests, as a replica node. This operation improves the network performance by reducing the response bottleneck, and by having a content distributed by the network, increasing availability and decreasing the latency.

There are several possible decisions about efficient content placement based on metrics, such as popularity and freshness. Replacement algorithms and strategies have a direct impact on the network's performance due to the increase of content availability. However, proper distribution should be considered to avoid high redundancy.

Content-based security

Security is applied to the content itself rather than the communication process. Different trust models have been developed based on network devices. Also, each data packet is self-authenticating based on the original contents security-related information, such as the Publisher public/secret keys signature.

Mobility

When a node moves from a network to another, it can re-issue any unsatisfied requests, and the Producer replies with the requested data without any need to request a new address during the movement [2].

2.2.3 Overview of ICN proposed architectures

Several ICN architectures have been proposed to address problems and limitations of host-centric models. In the following, some representative ICN architectures are briefly described: Data Oriented Network (DONA), Publish-Subscribe Internet Routing Paradigm (PURSUIT), *Mobility First*, Content Centric Networks (CCN) and NDN.

Data-oriented network architecture

One of the first ICN architectures, DONA [25], uses persistent flat names to identify information objects, instead of hierarchical Uniform Resource Locator (URL). It allows them to be cached and replicated at the network layer. It supports on-path and off-path caching, and also follows an early-binding approach, which means that content providers must register *identifier-locator* mappings before delivering a content [2]. The content Publisher uses a cryptographic hash as an object identifier. The content integrity is verified by the Consumer, by hashing it and comparing the results.

Publish-subscribe Internet technology

PURSUIT [26] designs an ICN architecture by using a publish-subscribe stack, instead of IP protocol stack. It uses persistent flat names as the name scheme of the content, the same as the proposed in DONA. Dedicated entities, named as *Data sources*, periodically refresh content publication states and maintain information regarding the CP and the Subscriber, by means of a pair of identifications from both entities. The architecture depends on a *Rendezvous* system for consumers to subscribe to advertised contents. This entity is responsible to match the subscription request with the data source, according to a unique name. Once it happens, the Data Sources forward new *Data* to the Subscriber.

Mobility First

This project mainly focuses on the mobility issue. For this architecture, entities attached to the network are identified by globally unique names, which are separated from their network addresses. It allows good mobility support by means of dynamic address bindings, where the name scheme is flat and self-certifying.

A Global Name Resolution System is in charge to manage all dynamic mapping between network address and object identifiers, as well as making sure these values are updated when a mobile node changes its network association. These names can be translated into one or more network addresses at various points in the network. A hybrid routing protocol is used, performing decisions based on both unique identifiers and network addresses, in a hop-by-hop basis. It supports on-path and off-path caching.

Content-Centric Networking

In CCN [27] data is published at nodes, and routing protocols are used for information distribution about the location of the data. The names are hierarchical, allowing name resolution and data routing information to be aggregated across similar names.

CCN supports on-path and off-path caching, following a Leave Copy Everywhere (LCE) strategy. It also follows a Listen First Broadcast Later (LFBL) approach, which is topology-agnostic and uses broadcast communication for all packets in the request phase, when it is assumed that the requester does not contain knowledge in advance.

Named Data Networking

Currently, NDN is one of the most active relevant architectures in ICN research. It is a derivative of a CCN architecture to support other needs unveiled upon further investigation.

Two packets are introduced: *Interest* and *Data*. Both packets carry the name of the requested content. Thus, NDN is a named-based network where routing is achieved by using names. They are hierarchical, human-readable, and structured URL like names. Moreover, the architecture maintains three data structures: Content Store (CS), Pending Interest Table (PIT) and Forwarding Information Base (FIB). These three components are integral parts of the NDN Forwarding Engine Module of a node, represented in Figure 2.4.

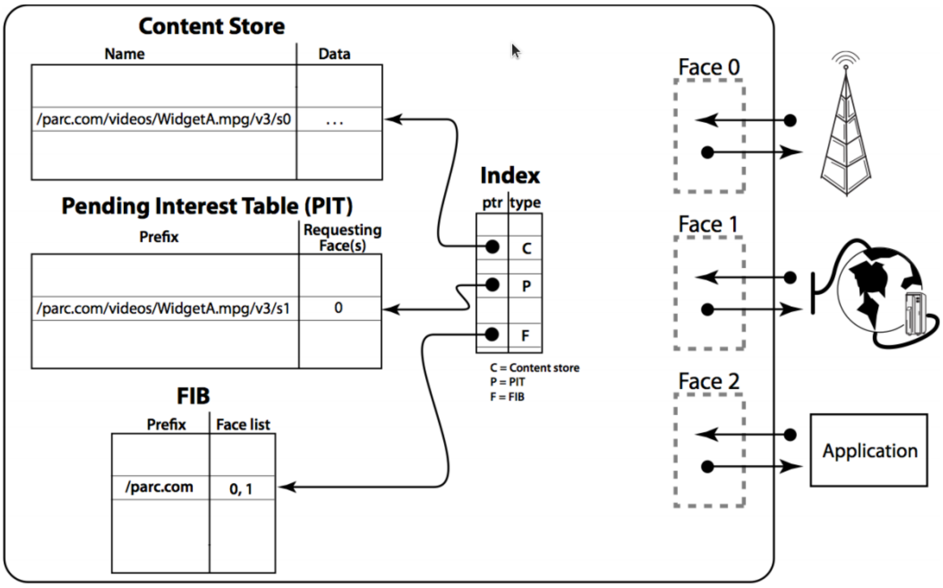


Figure 2.4: NDN Forwarding Engine Module [28].

Once a node receives an *Interest*, it checks its local CS. If the content is cached, it means that the node is a replica node, thus a *Data* packet will be forwarded to the same *face* where the *Interest* has been received. Otherwise, it appends the *face* where the *Interest* has been received to the PIT entry. Then, it checks the FIB to find the *face* towards the content.

The *Data* is assembled and sent to the reverse path following the *faces* which its associated *Interests* traversed, recurring to the *faces* pointed out by the traversed nodes' PIT. The nodes receiving the *Data* may cache the content.

NDN is the base network of choice for this work, as a promising candidate for the Future Internet [29]. It directly deals with application generated variable-length, location-independent names to pull contents for end-user, irrespective of their hosting entity. The data comes as a first-class entity. It secures the content and provides essential context for security, decoupling the trust of the content to the trust in hosts. Moreover, it has the benefit to use scalable communication mechanisms, for example: automatic caching to optimize bandwidth, and the potential to move content along multiple paths to the destination [30].

2.3 NDN & Consumer mobility

In NDN, the service for Consumer mobility is easily provided by the retransmission of an *Interest* for the content that is inquired but not received due to handover. Despite the simplicity of how the mobility concern is handled, there are drawbacks [31]:

- A new request and forwarding of the same content has to be established before handover, causing redundant *Data* transmissions.

- The retransmitted *Interest* has to follow a new communication path until it reaches the original or replica content. In general, the NDN node which has requested the content more than once due to having coverage from a different AP is at the intersection of the new and old *Interest's* path. Whether the route taken by the *Interest* intersects or not, the cache functionality of NDN may not be used effectively. Hence, new *Interest* turns NDN states, such as in-network caching and PIT entry, useless.

2.4 Related work

2.4.1 Consumer mobility in a native NDN environment

The issues raised in Section 2.3 are well known by the NDN research community, leading to proposals which aim to enable seamless ways to recover the communication damaged due to the mobility of end-users.

Mobility First ICN architecture belongs to this related work, since one of its design goals is to have seamless host and network mobility [32]. This solution has been detailed in Section 2.2.3; however, it is not NDN-based.

In [33], a proactive caching approach lets a candidate Network Access Router (NAR), that is expected to communicate with a Consumer after handover, to proactively retrieve and cache contents. NAR has additional functionalities when compared to a regular AP: it pre-fetches and caches content on behalf of the user. Once a mobile Consumer detects potential handover (detected by a stronger signal strength coming from another NAR), it communicates to its Previous-NAR (P-NAR) with a *Control Interest* packet. Once this happens, P-NAR stops transmitting the requested content by the user, despite still receiving and caching the content. Moreover, it sends a *Control Interest* to the Consumer's Next-NAR (N-NAR) which updates FIB entries on the way. The N-NAR starts pre-fetching the Consumer's content of interest from the ones cached at P-NAR the moment it receives the mentioned *Control Interest*. Later, the user fetches the unsatisfied content from the N-NAR due to handover interruptions, using normal *Interest* and *Data* packets. This proposals focuses on how to fetch the lost content again. It is also based on reissuing a new *Interest* through a new NAR, so that the problems of the existing NDN Consumer mobility scheme may be repeated.

The work in [34] is a proxy-based mobility management scheme in NDN to reduce the packet loss during the handover. It is used a centralized manager node to keep track of the NDN conditions, and each mobile node is associated with a proxy from the infrastructure. The IP address is used to exchange the identity between the mobile node device and the proxy, securing association between both parties. When a node hands-off from one proxy node to another, the new one retrieves the content cached at the previous proxy node, and caches at its cache. This approach is similar to the one developed in [33]. With the use of *Hold request* messages, the previous proxy node should be notified of imminent migration to prevent unnecessary packet loss and resource consumption. Then, after handover, the Consumer notifies its proxy node about its new IP address, with a *Handover notification* message, containing the content sequence numbers received at the old location. This proxy then transmits the requested content that was stored for the Consumer towards the new location of the mobile device. This proxy-based mobility management scheme reduces the packet loss during the handover. However, these solutions cannot optimally use on-path caching, and the packet transmission path is not optimized. Moreover, it needs exchange control messages between proxy servers with the help of the current IP network. Apart from that, it suffers from the same drawbacks mentioned in [33].

In [31], Mobile Link Service (MLS) supports Consumer mobility service as well as the content fragmentation and re-transmission function with the following two message types:

- MLS fragment message: the NDN packets are divided with the sequence number. MLS provides the following three main functions: a connection reestablishment, *Data* retransmission, and NDN packet fragmentation and reassembly.
- Status Report Message: request reestablishment of the connection and re-transmission of a lost NDN packet.

MLS uses a status report message to inform the previous NAR about the status of the transaction, such as the connection information that is changed after the handover of the Mobile Node Consumer (MNC) and employed to request re-transmission of a MLS fragment message. For every received *Interest*, a new transaction is made at the NAR. The *Interest* is passed to an upper NDN forwarding plane. When receiving a *Data* packet from the NDN layer as response of the *Interest*, MLS fragments these packets into several MLS fragment messages, which are sent to the MNC. The MNC sends the status report message that includes the new transport endpoint information and the sequence number of the missed MLS fragment message to the previous NAR due to handover. Upon the receipt of the status report message, the NAR updates the address information of the transaction structure and re-transmits the lost MLS fragment messages. The proposed solution does not keep track of the condition of a connection for a transaction, thus in the end it cannot handle the movement of MNC, but only the recovery of lost packets from the previous NAR.

Some of these proposals focus on how to fetch the lost content again. And some of these schemes are also based on reissuing a new *Interest*. Hence, the existing problems of the NDN Consumer mobility scheme may be repeated. Moreover, some of the solutions resort to IP as an overlay to achieve seamless connectivity, unfulfilling ICN principles. Finally, most of the described related work focuses on the recovery of packets from the previous AP (before handover) through support packets, and not the direct reception of the content from the CP or replica node to the new AP where the Consumer is located. The work covered by this dissertation focuses on the deviation of the *Data* packet from the CP or replica node to the AP where the Consumer is to hand-off to.

2.4.2 Consumer mobility in an NDN pub-sub environment

Several ICN-based solutions have been proposed to implement the publish-subscribe model; however, there is a lack of solutions tailored to support Consumer mobility scenarios.

The work in [35] has a publish-subscribe scheme based on [36]. It handles asynchronous communications between ICN entities. The Publisher provides information, and then the

Subscriber may request this information and consume it at any time. After a movement, the mobile node just has to reissue its lost *Interest*, and then the network directs that request to nearby CSs or original Producer [37]. The Consumer assigns a timeout to each pending request. When the timeout expires, the same request is sent again to the Producer.

The work presented in [38] proposes a pull-based approach that adds broker nodes to the original architecture. The 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 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 [39]. The Subscribers are required to send the *Interest* packet periodically (heartbeat *Interest*). This heartbeat provides more flexibility to the broker, working as a Consumer registration to the broker. For example: it is able to pause the *Data* transmission if the heartbeats stops. If the Subscriber has moved from its original place, it will resend another heartbeat *Interest*. The heartbeat *Interest* will be satisfied by one of the upper layer routers, due to in-network caching, or by the broker itself.

The work in [40] introduces a new network entity (Rendezvouz Point), two new messages (Subscribe and Publish), and a new table (Subscription Table). In this case, the Consumer interested in subscribing to a specific topic, issues a *Subscribe* message to its local Rendezvous Point. This message is forwarded towards the reference Rendezvous Point in the network, according to the information stored in FIBs of intermediate routers. During this process, involved nodes populate their Subscription Table with pending subscriptions. When a content is generated, it is sent within a *Publish* message to the reference Rendezvous Point. Then, this message is delivered to all the subscribed Consumers, according to information stored in the Subscription Tables. The subscriptions are handled with subscription renewals through validity periods. A validity time assigned to a subscribe message defines how long the Consumer is interested in receiving publications from the Rendezvous Point.

To ensure the reception of messages lost by connectivity loss, the Consumer renews the subscription detailing information for retrieval of the lost content. If the message arrives to a router where a similar subscription already exists, then it is not further forwarded. It checks if it has the next publications in its local cache. If they exist, it resubmits all the pending publish messages towards the Subscriber. If the node does not find the message in its cache, it sends an *Interest* to a Back up Node (BN). BNs are subscribed to all the content descriptors in order to store all the events and be able to provide these contents. They can be dedicated a server or a cluster of servers required for better scalability and fault-tolerance system.

2.5 Mobility forecasting

With the popularity of 5G and IoT, it is expected a demand growth on the mobility support in the network. For future networks to overcome current limitations imposed by the mobility of end-users, more intelligence is needed for a fully autonomous and flexible network. Non-smooth handover may bring on unsuitable delays and call dropping events, which needs proper handover management [41]. An efficient way to manage mobility in order to maintain user's continuous connections is by resorting to mobility prediction [42]. Accurate predictions of user mobility would provide efficient resource and handover management, providing them better quality of service [43]. Many important works have been dedicated to this issue: based on Markov chain, hidden Markov model, Artificial Neural Networks, Bayesian Network and others.

For instance, the work in [44] achieves a low overhead solution to support the Consumer's mobility, by using geolocation vehicle mobility forecasting modules, which estimate the future connected RSUs. It uses a scheme to forecast the content request distribution, based on the estimation of content popularity. It implements a learning-based algorithm to proactively cache the user requested content at the RSUs. This work is dedicated to Over the Top (OTT) networks.

None of the mentioned related work takes advantage of available intelligent entities to handle content transmission to the new Consumer's location.

2.6 Chapter considerations

With the advancement in the trends towards 5G and increasing popularity of IoT, mobility has become a norm. This chapter introduced the resulting different mobile architectures, including VANET. Moreover, it introduced the issues induced by the Consumer's handover in current network architectures.

This chapter presented ICN as an approach to evolve the Internet infrastructure to overcome the drawbacks of IP, including the ones related to mobility. ICN uses the content name, which is unique, persistent and location-independent, as the pillar of the entire communication flow, enabling in-network caching and content replication, therefore facilitating efficient and timely delivery of information. Key features were described, as well as the most relevant ICN-based architectures, including NDN, which is used in this work. Concerning NDN, we explicitly describe how it deals with the Consumers mobility and what are its core issues.

Taking this into account, research and development has been made to achieve recovery of lost packets due to handover, diminishing intermittent communication during this transaction. However, some of them do not fully comply with ICN principles, by still recurring to IP, and most of them only implement *Interest* retransmission. Therefore, the mobility problems are still able to persist.

Relevant mobile forecasting modules have been used in the research to add intelligence to the network infrastructure. As explained in this chapter, these were introduced to achieve a more efficient manner to sustain continuous connections throughout the mobile Consumer's trajectory.

Knowing the efficiency of forecasting algorithms to calculate the next positions of mobile devices and their sustainability in IP network infrastructures by applying new delivery schemes, we will propose in the next Chapter a new NDN-based scheme to support efficient seamless connectivity to the consumers in situations of hand-offs, for both single content requests and publish-subscribe communication model, by resorting to a mobility prediction algorithm hosted by a remote machine.

CHAPTER 3

NDN-based Scheme for Seamless Connectivity

This chapter presents solutions that empower the seamless delivery of contents to mobile devices, in an NDN based network. First, this chapter overviews the base network and communication protocols available in the beginning of this work. Then, two mobility-based content delivery approaches are proposed, each one regarding a different communication paradigm: native NDN, denoted as Single-Data-Request throughout the remaining document, and publish-subscribe, simply denoted as pub-sub. We start by introducing the problems of the native solution when there are mobile Consumers involved in the network topology. Afterwards, we propose an approach that is able to overcome these problems. We then present the entities, packets, mechanisms and interactions involved at the NDN-based network to address a seamless mobility approach.

3.1 Base work overview

The proposed architecture is conceived taking as a basis the NDN architecture, explained in Section 3.1.1. It takes advantage of a routing protocol, denoted as Named Data Link State Routing Protocol (NLSR) and is described in Section 3.1.3, and a pub-sub model is described in Section 3.1.2.

3.1.1 Named Data Network

The proposed architecture is developed using a pure NDN [45] architecture as a base model. This architecture employs two types of packets: the *Interest* packet, to request content, and the *Data* packet, to deliver it. Both of them have a common content identifier denoted as *Name*. It is represented by a sequence of *Name* components, following a hierarchical structure with variable length.

Named Data Network Link Protocol (NDNLP)

NDN uses a link protocol known as NDNLP [46]. It is a single-hop protocol responsible for message encapsulation and exchange. It introduces the following features:

- *Fragmentation and Reassembly*: the sender fragments the message into multiple packets, respecting the imposed Maximum Transmission Unit (MTU). It is reassembled into the original message by the receiver.
- *Negative Acknowledgment and retransmission*: indication of unsatisfied *Interest* packets through the downstream transmission of Negative Acknowledgement (NACK) packets. It notifies the inability of the forwarder to satisfy a particular *Interest*. It may reveal the reasons, such as congestion on the link, detection of duplicate *Nonce*, and no route matching the *Interest*.
- *Acknowledgment and retransmission*: the sender retransmits lost packets. The receiver acknowledges the reception of packets in a summary bitmap.
- *Consumer-Controlled forwarding*: allows an application to indicate the outgoing *face* of an *Interest* packet.
- *Cache Policy Control*: a Producer can indicate how its *Data* should be cached or not.
- *Integrity*: prevents packet injection.

NDNLP is designed to be of use by all types of connections, including UNIX sockets, Ethernet, User Datagram Protocol (UDP) & TCP connections and WebSockets. It operates as a link adaptation layer, located between link and network layers.

In NDNLPv2, both *Interest* and *Data* packets are interpreted as *LpPacket*. This packet can contain additional information, represented as *tags* encoded into the header field. The forwarding plane is not forced to preserve *tags*. Nonetheless, it becomes beneficial in cases where it is necessary to update these parameters along a path.

NDN node

In the context of NDN, all devices that exchange information, according to the NDN logic, are designated as nodes. A node is able to perform three types of functionalities: Producer, assembling and signing contents into *Data* packets; Consumer, issuing *Interest* for specific contents; Router, conveying *Interest* and *Data* packets.

Each NDN node manages three structures: CS, PIT and FIB. The CS caches the content of incoming *Data* packets; it allows for a faster content retrieval specified by the *Name* in the incoming *Interest*. PIT is a table mapping the *Name* of the requested content along with the node's interface, known as *face* in NDN context, from which the *Interest* has entered. This entry represents a pending content request. It is stated as satisfied only when the correspondent *Data* reaches the node. Once satisfied, it is removed from the PIT. Finally, FIB is a table which records contain the *Name* prefixes of the content and respective outgoing *faces*. It allows routing of the *Interest* through a name-based lookup. The component known as *faces* enables to receive and forward packets; thus, it behaves as an interface with appropriate mechanisms.

When an *Interest* is sent to the network, it is dispatched with resort to the FIB table. This is done successively until it reaches a node containing the queried content. Each intersected node by the *Interest* will check the availability of a fresh queried content in the CS. If the node hosts such content, whether it is an intermediate node or the Producer itself, it will reply by sending a *Data* packet containing the requested data. The message will follow the reverse path taken by the *Interest*. Consequently, it will arrive to the respective Consumer. This is done by pursuing the mentioned PIT entries of the traversed nodes. Figure 3.1 illustrates the flow of the *Interest* and *Data* packets, such as the mechanism imposed by the node it traverses.

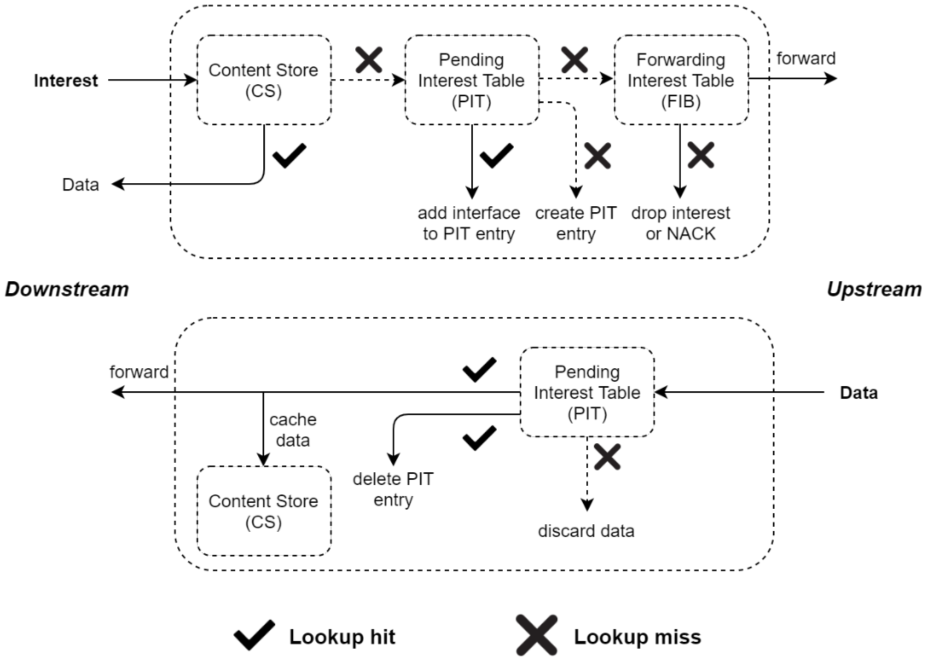


Figure 3.1: Packet flow from an NDN network node's perspective [47]

Interest NACK are used for notifying downstream nodes about forwarding failures or requests for incorrect or unavailable information. The messages flush state in the routers and notify the consumers. They have other purposes, such as helping to mitigate *Interest* Flooding attacks [48]. These messages are transmitted just as the *Data* packet, by following PIT entries.

Interest

NDN *Interest* is a packet defined by a Type Length Value (TLV) format. *Name* and *Nonce* are the only required parameters in the packet. The first identifies the requested content. It has at least one *Name* component, otherwise the packet will be discarded by an NDN node. The second uniquely identifies the message. It is used to detect looping *Interests*, which are important to be avoided.

The optional parameters of the packet are the following:

- *CanBePrefix*: when present, the *Name* value is a prefix of the full *Name* of the requested *Data* packet. Otherwise, it is either the exact or full *Name* of the *Data* packet.
- *MustBeFresh*: Indicates whether the CS may satisfy the *Interest* with stale *Data* or not.

- *ForwardingHint*: Contains a list of *Name* delegations used by the forwarding logic.
- *InterestLifetime*: Indicates the remaining time before the *Interest* times out. The value is the number in milliseconds. The timeout is relative to the arrival time of the *Interest* at the current node.
- *HopLimit*: The optional *HopLimit* element indicates the number of hops the *Interest* is allowed to take.

The format of the *Interest* packet is depicted in Figure 3.2.

Data

NDN *Data* is a packet defined by a TLV format. The parameters are the following:

- *Name*: Identifies the content.
- *MetaInfo*: contains additional information about the *Data* packet itself. Contains three optional fields:
  - *ContentType*: identifies more specifically the content type. The following values are briefly explained in Table 3.1.

Table 3.1: ContentTypes details.

ContentType	**Assigned number**	**Description of the content**
BLOB	0	Payload identified by the *Data*name; This is the default *ContentType*
LINK	1	A list of delegations
KEY	2	Public key
NACK	3	Application-level NACK

  - *FreshnessPeriod*: indicates how long a node should consider the arrival of the content before marking it as *non-fresh*. The encoded value is the time in milliseconds. Note that the *non-fresh Data* is still valid data; the expiration of *FreshnessPeriod* only means that the Producer may have produced newer data.

    If an *Interest* contains *MustBeFresh* element, a node cannot return a *non-fresh Data* in response.

    When a duplicate of the *non-fresh Data* packet with positive *FreshnessPeriod* arrives at the node, it should remark the stored content as *fresh* for the specified duration.
  - *FinalBlockId*: indicates the final block in a sequence of fragments. It may also be present in other fragments to provide advanced warning of the end of the content to consumers.
- *Content*: contains the data itself.
- *DataSignature*: it is defined as two consecutive TLV elements: *SignatureInfo* and *SignatureValue*.
  - *SignatureInfo*: fully describes the used digital signature algorithm and any other information to locate its parent certificates.

– *SignatureValue*: holds the actual bits of the signature. It covers all TLV elements inside the *Data*, starting from *Name* and up to *SignatureValue*.

The format of the *Data* packet is depicted in Figure 3.2.

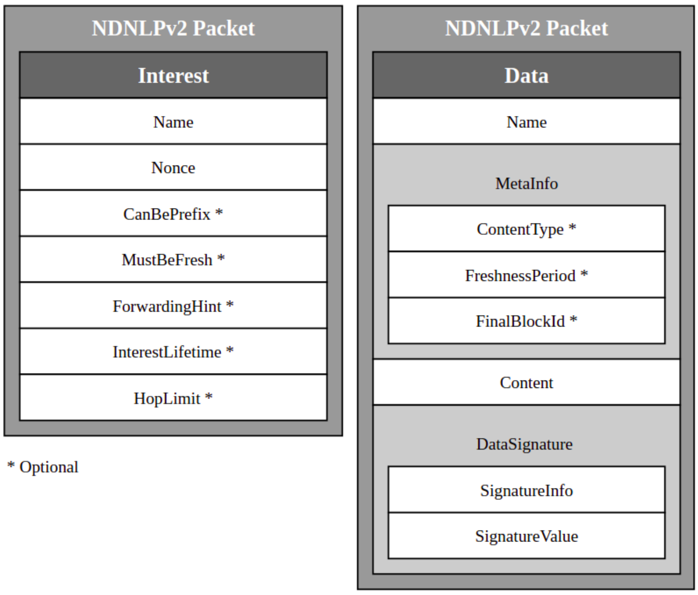


Figure 3.2: *Interest* and *Data* packet formats.

3.1.2 Pub-sub model

Apart from the NDN native communication, also denoted as the *Single-Data-Request* communication throughout this document, the proposed network takes advantage of a pub-sub communication model: forwarding multiple *Data* packets, throughout time, using a single *Interest*. Hence, fewer *Interest* packets will be flowing through the network in a pub-sub model. There is a subscription to a type of content, and when this type of content is available, it is proactively forwarded to the Consumer.

The work in [49] classifies this type of traffic in single-request/multiple-response, periodic delivery, n responses and conditional delivery. In single-request/multiple-response, a Subscriber sends one request asking for *Data* which may comprise of multiple responses spread over time. In periodic delivery, Consumers send one request packet asking for periodic *Data* identified by the name after a specific time interval. In n responses, Subscriber nodes send one request packet for a specific number of responses, and in conditional delivery, Subscribers send a request packet to receive *Data* from a Publisher only if certain conditions are met or events are triggered. It enables seamless pub-sub model, authentication, access control, and group management features, without modifying ICN principles. The authors argue that their strategies, based on semi-Persistent Interest, achieve lower control overhead, and that the memory requirements of the core nodes can be kept at minimal levels.

Most of the works implementing pub-sub for ICN use solutions based on *Persistent Interests*. Another example would be the work done in Information-Centric Approach to Secure Home Energy Management (iHEMS) [50]: it allows the NDN node to carry out persistent subscription, using long-lived forwarding information. The subscription request, sent by the subscriber, persists in the nodes' PIT during some configured time.

We make use of this behaviour in our approach: we use *Persistent Interests* in PIT entries created as a result of an incoming *Interest* packet, marked as a subscription. The mark is done with a specific *LpPacket* tag.

The process of handling *Persistent Interests* is formalized in Algorithm 1. The value denoted as *InterestLifetime* is employed to be the lifetime of the persistent time of the PIT entry (line 2). To a specific content, if a *Persistent Interest* has a greater *Interest* lifetime value than the remaining time of the respective persistent PIT entry (line 7), then the pointed value is considered (line 8). Otherwise, the lifetime of the PIT persists until timeout or another update.

Algorithm 1: Pit Entry Lifetime Decision

```
1  Algorithm ProcessPersistentInterest(face inFace, PersistentInterest I)
2  InterestLifetime = getInterestLifetime(I)
3  InterestName = getName(InterestName)
4  PitEntry = getPitEntry(inFace,Iname)
5  if PitEntry != null then
6  |   PitEntryLifetime = getPitEntryLifetime(PitEntry)
7  |   if PitEntryLifetime < InterestLifetime then
8  |   |   update Persistent Pit Entry Lifetime with InterestLifetime
9  |   end
10 else
11 |   createPitEntry(inFace,I)
12 end
```

3.1.3 Named Data Link State Routing Protocol

NLSR [51] is a routing protocol to populate NDN RIB. It sets up the routing table, using link-state with the purpose of disseminating both topology and *Name* prefix information, instead of IP prefixes. For each reachable *Name* prefix, it produces multiple faces in a single authoritative domain. NLSR is evolving over time to become a full fledged inter-domain routing protocol for NDN.

The naming scheme designed by NLSR indicates the relationship among various entities in a routing system. It has developed a trust model for key verification of a routing protocol, and made adjustments to NDN's new design patterns of using *Interest-Data* exchanges to spread routing update messages.

Since the protocol works with NDN's *Interest* and *Data* packets to propagate routing updates, it focuses in terms of *Data* names and *Data* retrieval. Authors of [51] proposed a systematic naming scheme for routing and routing updates, which are retrieved without prior knowledge of the time an update may be generated, since topology changes may happen in the course of time.

This protocol distinguishes itself as a link-state routing protocol: it provides multiple next hops for each *Name* prefix, and verifies and signs all Link State Advertisement (LSA) messages, ensuring that each router is able to originate only its own prefix and connectivity information.

3.2 Proposed solutions for seamless connectivity

In this section we introduce new mechanisms, packets and interactions to handle seamless connectivity for mobile consumers. Two solutions are discussed, each one for a different communication type: *Single-Data-Request* in Section 3.2.1, and *pub-sub* in Section 3.2.2.

Before proceeding to the next sections, one has to notice that the referenced topologies, apart from hosting Routers, Producers and Mobile Consumers, include Road Side Units (RSUs)/Access Points (APs).

As stated in [52], RSUs are network elements in the road that work as traffic directories, location servers, data disseminators, security managers and service proxies. They come up as a solution to allow OBU and other devices, with wireless communication interfaces, to connect to the Internet and with other moving devices.

The RSUs from the wired network take advantage of NLSR to disseminate its configured *Name* prefix and topology, throughout the network. Having the topology advertised, each node is capable to know which *face* will lead, with a minimum number of next hops, to a specific RSU, by knowing its configured *Name* prefix, associated at the RIB table. Figure 3.3 shows an example of configured RIB table of a node, from a small topology, according to what has been mentioned. By the information given by the RIB, it is possible to create a forwarding method recurring by this table, sending the content towards a specific node. In Section 3.2.1, a node is then capable to transmit a new *Data* Support packet, named *Data Switch* (DS) packet, to specific RSUs, resorting to the RIB table.

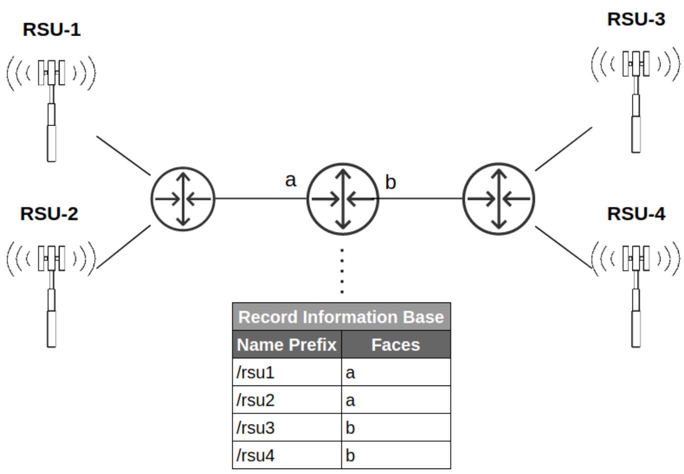


Figure 3.3: Example of a configured RIB table.

3.2.1 *Single-Data-Request*

Routes taken by the mobile consumers might originate handovers. Consequently, the settled connection might get interrupted throughout the process. Consider a mobile node, which is about to switch to another AP, sending an *Interest* packet, as depicted in Figure 3.4: because of topological factors, such as link delay, queue delay, computing process delay and other circumstances, transmission of *Interest* and *Data* packets might not be fast enough, hence the Round Trip Time is not short enough for the mobile Consumer to receive the content at the AP/RSU which it has been previously connected to. So, in the case of Figure 3.4, the car issues the *Interest* at RSU-1. However, by the time the content reaches the RSU-1, the Consumer is no longer in reach from that AP.

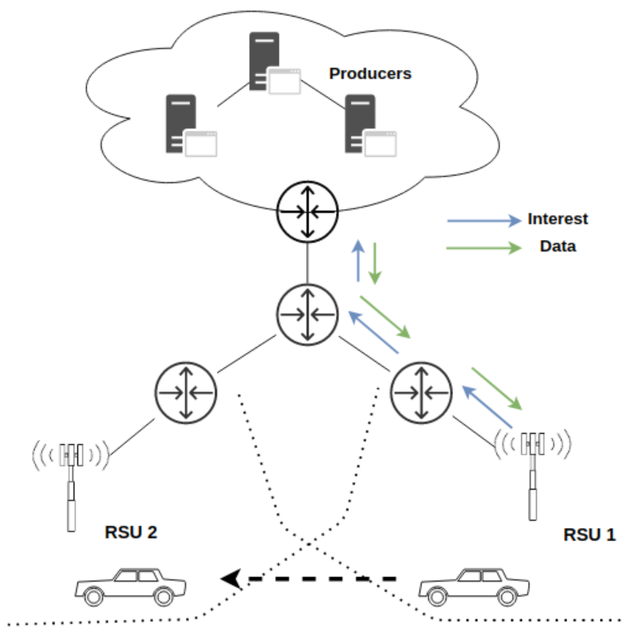


Figure 3.4: Example of unsatisfied *Interest* caused by mobility of the Consumer.

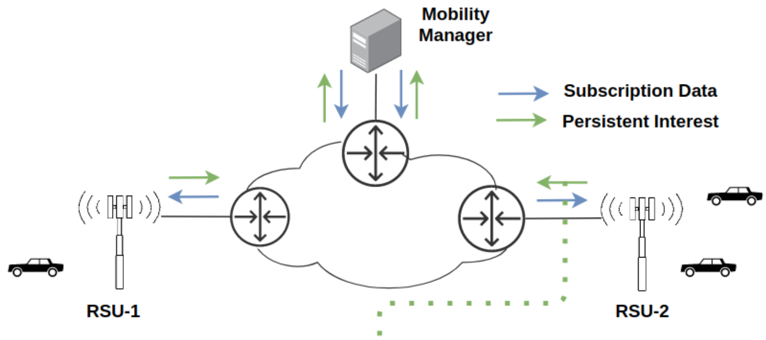
To address this problem, ideally the content has to be deviated to the RSU which belongs to the mobile node's next position. Considering the possible RSUs a mobile Consumer will most likely traverse, and the instant the mobile node enters its respective transmission area, it is possible to transmit the traffic to the new location of the Consumer. To this extent, the network has to adapt itself when such conditions are determined, so the content might get deviated to the prospective RSUs, hosting the mobile Consumer. In the example represented in Figure 3.4, it is aimed for the *Data* packet to traverse the RSU-2, so that it can reach the Consumer at its new position. Such solution can prevent loss of data packets while the movement occurs.

Beyond elements such as Producer/Publisher, Consumer/Subscriber, AP/RSUs and intermediate nodes (backend routers), the infrastructure must host an additional element: a Mobility Manager (MM).

Forecasting next mobile positions

The MM is a node that hosts a forecasting module used to calculate and communicate potential migrations of the infrastructure's mobile devices. By migration, we mean transition of a mobile Consumer from their current RSU to another, denoted as *next RSU* throughout the remaining document. In order to forecast the next position, the MM periodically receives knowledge about the recent state of the different mobile devices, and provides interchanging packets with the RSUs.

These mentioned mobile consumers states are issued from the RSUs using pub-sub communication. They act as a Publisher of such information, which is to be provided to the MM, that behaves as a Subscriber.



Data Subscription		
Name Prefix	**Description Content**	**Subscription Tag**
/rsu1	[{mac: f8:ac:65:ba:5e:eb, rssi:-50db}, {mac: 02:42:6c:37:7f:30, rssi: -60db}]	TRUE

Figure 3.5: Communication for RSU's neighbors status.

To that end, each RSU has configured a unique content topic dedicated to the transmission of *Data* packets containing information about its mobile neighbours (such as Media Access Control (MAC) addresses and the received Received signal strength indication (RSSI), obtained from the resulting beacons operating in the lower layer), which is subscribed by the MM. In Figure 3.5, the MM subscribes to two RSUs by transmitting two *Persistent Interests*, which name prefix of the topic corresponds to the configured Name of the RSUs. For example, the topic Name of RSU-1 is */rsu1*. Moreover, we can see that the *Data* packets sent by RSU-1 host information related to its neighbors (MAC address and RSSI). This data is transmitted to the topic already mentioned. The MM uses this information to forecast the next positions of mobile nodes and help our solution to provide seamless communication for mobile consumers.

We present a complete solution with the supportive services to host a mobility predictor, following the NDN paradigm. The responsibility for detecting potential migrations of mobile nodes between RSUs is exclusive to the MM. We highlight that this work assumes the existence of a mobility predictor service, as the ones presented in [53], as this is not the contribution of this work.

Prediction request

We present, in Algorithm 2, the steps taken by an RSU after receiving an *Interest* packet through the wireless interface, sent by a mobile a Consumers .

By default, once an RSU receives an *Interest* wirelessly from a querier, it searches at the CS for the requested content (lines 2-3). In case it is cached at the RSU, it is wirelessly sent to the mobile Consumer in the format of a *Data* packet (lines 4-6). Otherwise, the RSU communicates with the MM module, as observed in Figure 3.6, through another *Interest* packet (packet number 1), representing a prediction query (lines 10-11).

Algorithm 2: RSU processing method triggered by the reception of Consumer *Interests*.

```
1  Algorithm processInInterest(Interest I, String ConsumerMAC, face inFace)
2  ContentName = getInterestName(I)
3  Content = checkConctentStore(ContentName)
4  if Content != null then
5  |   DataPkt = Data(ContentName, Content)
6  |   sendPacket(DataPkt, inFace)
7  else
8  |   SequenceNumber = createSequenceNumber()
9  |   storeInterestPkt(I, SequenceNumber)
10 |   InterestPrediction = createPredictionQuery(ConsumerMAC, SequenceNumber)
11 |   sendPredictionQueryPacket(InterestPrediction);
12 end
```

It is allocated a sequence number to this packet, which is appended to the Name prefix of the message as a name component. The later is associated with the *Interest* received from the Consumer, additionally stored at the entity for a short period of time (lines 8-10).

The name-prefix contains the following components:

- Identification of MM: selects which MM from the topology should the forecasting results be calculated and transmitted from.
- Action: specifies the type of content desired by the requester.
- Identification of the RSU: the RSU hosting the mobile inquirer.
- Identification of the Mobile Consumer: the Consumer associated to the predicted *nextRsus*.
- Sequence Number: associated with the *Interest* packet requested by the Consumer.

An example of a name of this packet would be:

/mm1/forecasting/rsu3/f8:ac:65:ba:5e:eb/%FE%01

More variable name components can be thought to satisfy the MM needs. This one gives the required instances to determine potential *next RSUs* of a mobile Consumer.

Once the MM receives the aforementioned packet, the following happens. First, it checks if the aforementioned name prefix structure is according to what has been predetermined. Additionally, it checks if the first name component of the prefix equals its own identifier. If both conditions are correct, it retrieves the name of the component identifying the Consumer by its MAC address in order to determine its potential *next RSUs*. Otherwise, no reply is given by this MM. Finally, it creates and sends a *Data* packet, containing the name prefixes of the *next RSUs*. Moreover, the name prefix of the packet is equal to the one designated to the received *Interest* packet.

The RSU needs fresh forecasting results. If predictions for a specific Consumer are outdated, then the actions taken by the hostess RSU might not be the appropriate ones.

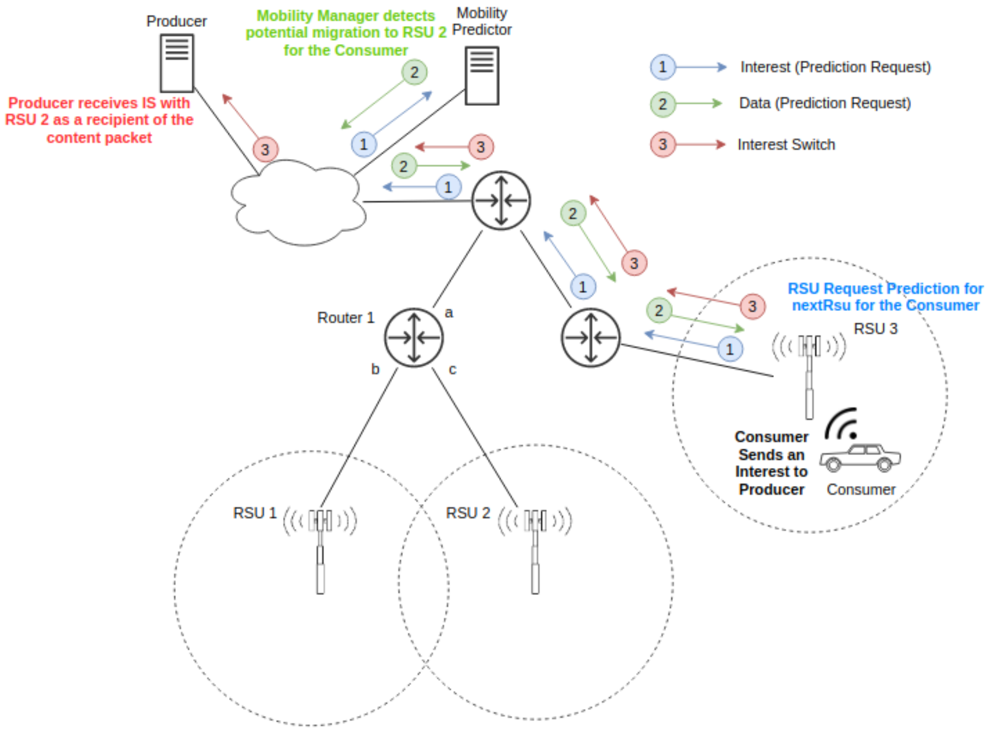


Figure 3.6: Example of a subsequent transmission of *Interest* with prediction request from the MM (1), the *Data* packet containing the predictions (2) and the consequent *Interest Switch* (IS) (3).

Notice that the content provided by the aforementioned *Data* packet and its Name prefix is not to be stored at the CS at the intermediate nodes it traverses. This forces the *Interest* inquired from the RSU to reach out the MM for recent information calculated at the correspondent MM. For this to occur, the *Interest* representing the forecasting queries has to indicate at the field *MustBeFresh* that the *Interest* is only satisfied with stale Data. Besides this, the *FreshnessPeriod* indicated by the *Data* packet must be *null*.

RSU decision mechanism

This section presents the Algorithm 3, formalizing the process triggered by the reception of the prediction information about one of the neighbors the RSU is hosting.

When an RSU receives a *Data* packet containing the forecasting results calculated only by the MM, it retrieves the sequence number from the name prefix (line 3), and then it searches for the temporarily stored *Interest* associated with the sequence number in consideration (line 4). The RSU selects the designated *next RSUs*.

Knowing the next position of a mobile node and proactively preparing the *next RSUs* for a seamless communication, it is now possible to prepare the new path between the Producer and the new position of the Consumer. To perform these actions, two new types of packets were introduced: *Interest Switch* (IS) and *Data Switch* (DS).

Algorithm 3: RSU decision triggered by the received prediction content.

```
1  Algorithm processPredictionContent(Data D)
2  Name = getName(D)
3  SequenceNumber = getSequenceNumber(D)
4  interest = extractInterest(SequenceNumber)
5  Content = getContent(D)
6  NextRsus = getNextRsus(Content)
7  if length(NextRsus)>1 || (length(NextRsus)==1 && !ContainsThisRsu(NextRsus)) then
8      SendBack = ContainsThisRsu(NextRsus)
9      interestSwitch = createInterestSwitch(interest, NextRsus)
10     sendInterest(interestSwitch)
11 else
12     sendInterest(interest)
13 end
```

Support packets

- **IS**: it is a derivative of the *Interest* packet with fields specifying the recipients. It follows the FIB entries of the intermediate nodes.
- **DS**: it is a derivative of the *Data* packet, containing fields specifying the new recipients. It follows both PIT entries and RIB entries. The consequent processing of the packet will later be detailed in this section.

Both IS and DS contain new fields:

- *NumRecipients*: specifies the number of recipients of the consequent DS packet. This field helps decoding the packet, so it can interpret correctly the *Name* prefixes of the subsequent DS recipients.
- *RecipientPrefix*: represents the *Name* prefix of the recipient of the subsequent DS. The number of *RecipientPrefix* fields added to the packet, specifying each recipient, is equal to the number specified in the field *NumRecipients*.

The IS has an additional field, called *SendBack*, a bit field that denotes if the content is to be additionally sent to the path it originally went through. In case the flag is disabled, no effective PIT entries are built. Hence, the content will not be transmitted to the nodes' incoming faces from which the respective *Interest* has traversed, if no other factor, other than the PIT, does not assign such route. The structure of these new packets is illustrated in Figure 3.7.

If the content is to be disseminated to other RSUs, then an IS must be created (lines 7-11 in Algorithm 3), based on the prior *Interest*, adding the correspondent information in the fields: *SendBack*, *NumRecipients* and *RecipientPrefix*. The *SendBack* flag is enabled in the case the content is to be additionally sent to the current RSU (line 9). If it is decided that migration from a node is unlikely, then it is sent the regular *Interest* packet earlier cached (line 13).

In Figure 3.6, RSU 3 realizes from the MM that the neighbor Consumer is most likely to travel to the transmission area from RSU 2. Hence, it composes an IS as explained before.

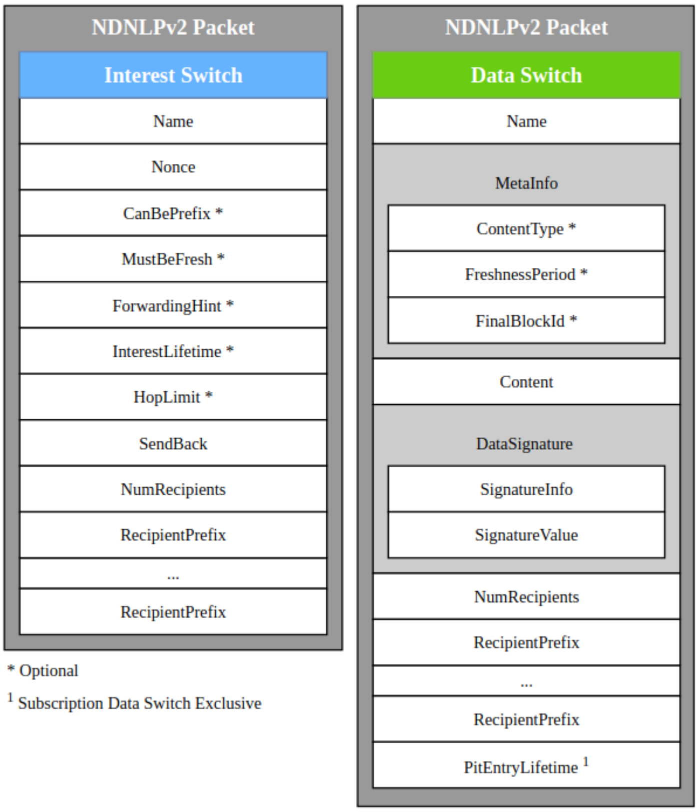


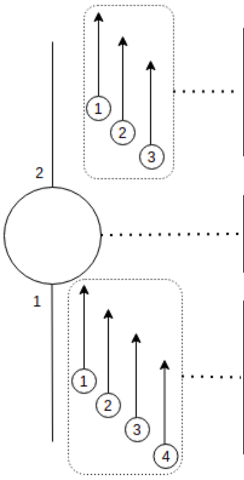
Figure 3.7: Structure of the support packets introduced for both solutions.

Interest Switch forwarding

The mechanism to process and transmit an IS is very similar to the regular *Interest* packet; however, it differs in the following:

- When *SendBack* is not enabled, no entry is given to the PIT. Therefore, the content will not circulate the reverse path taken by the IS;
- By default, when a regular *Interest* is received, the forwarding module will not route the packet in case there is already a PIT entry associated with the *Name* prefix denoted by the message. However, if the PIT entry is not yet satisfied and one or more ISs are introduced in the mean time, these cannot be ignored. This is introduced because, besides representing a request of content, it also represents the delivery of the respective data to other recipients. Hence, the packet cannot be thrown away, so its delivery reaches the denoted recipients;
- The node is programmed to register PIT entries, as explained in section 3.1.1, along with the *Name* prefixes of the recipients indicated by the IS. An IS is only thrown away if the *Name* prefix of the packet is associated with a PIT entry, and if the *Name* prefixes of the indicated recipients are registered on the same entry. Otherwise, either a PIT entry is created or updated with the *Name* prefixes of the new recipients. An example of the consequence of such mechanism is shown in Figure 3.8: the packet number 3 updates its *RecipientPrefix* fields, containing only the recipient with Name */rsu3*, since the other

recipients have been already specified by the *Interest* 2, which has the same prefix as packet 1, 3 and 4. Packet 4 is dropped, since no new recipients have been specified.



Packets			
Legend	Name Prefix	SendBack	Recipients
1 (Regular Interest)		-	-
2 (Interest Switch)	/prefix/content	FALSE	/rsu1,/rsu2
3 (Interest Switch)		FALSE	/rsu3

Pending Interest Record Table			
Name Prefix	Faces	SendBack	Recipients
/prefix/content	1	TRUE	/rsu1,/rsu2,/rsu3

Packets			
Legend	Name Prefix	SendBack	Recipients
1 (Regular Interest)		-	-
2 (Interest Switch)	/prefix/content	FALSE	/rsu1,/rsu2
3 (Interest Switch)		TRUE	/rsu2,/rsu3
4 (Interest Switch)		TRUE	/rsu1

Figure 3.8: Example of mechanism imposed to the subsequent *Interest* and *Interest Switch*, with equal *Name* prefix, when the associated PIT entry is not yet satisfied.

Data Switch dissemination

Once an *Interest Switch* is received and there is a content hit either at an intermediate node or Producer, it proceeds by composing and transmitting a *Data Switch*. Besides incorporating the requested content, the packet contains the Name prefixes of the content receivers, in TLV fields *RecipientPrefix*, specifying the total number of recipients in the field *NumRecipients*.

When a DS is about to bifurcate a new path (Figure 3.9), pointed exclusively by the RIB entries, it sends to the corresponding faces composing only the pointed RSU name prefixes by the RIB entries. The DS that still follows the PIT entries excludes the denoted recipients indicated by the mentioned DS, which causes a new bifurcation. If all necessary data switching are deployed throughout the transmission of the content, and no new recipients are indicated, then a *Data* packet is transmitted, behaving as natively.

In Figure 3.10, bifurcation of the content is caused by the DS. Instead of transmitting the content of packet 4 to the same path the inquiry has been made (in direction to the RSU 3, which path is shown in Figure 3.6), it follows the path to the recipient specified by the respective IS, resorting to the RIB of the intermediate nodes.

Once an RSU receives a DS, it checks if its prefix is one of the recipient prefixes in the packet. In that case, it proactively transmits the Data, as a regular *Data* Packet, to the wireless interfaces in order to increase the probability of content delivery to newly arrived mobile Consumers. This behaviour is also shown in Figure 3.10: once the RSU 2 receives the DS, specifying its configured name as recipient (packet 4), the same entity transmits proactively the content as a *Data* packet (packet 5) to its wireless interface.

Moreover, the content provided by the DS is stored at the CS of the considered RSU. The motive for this action comes as a last resort to provide a more efficient content forwarding: the

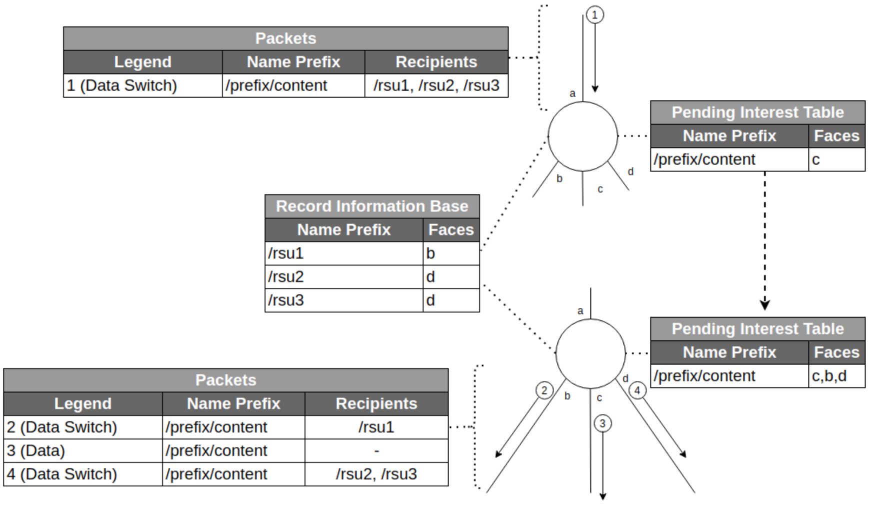


Figure 3.9: Example of content deviation caused by *Data Switch* bifurcation.

Consumer may have issued an interest for the content. However, the later was not delivered to the inquirer due to nonexistence coverage of its previous and next AP, when the *Data* packet was wirelessly broadcasted by the RSUs. If the Consumer is programmed to reissue the same unsatisfied Interest, once it has coverage with its new associated RSU, the cached content is directly forwarded from it. This is beneficial to improve the delivery of the content, enhancing the quality of experience for the end-users.

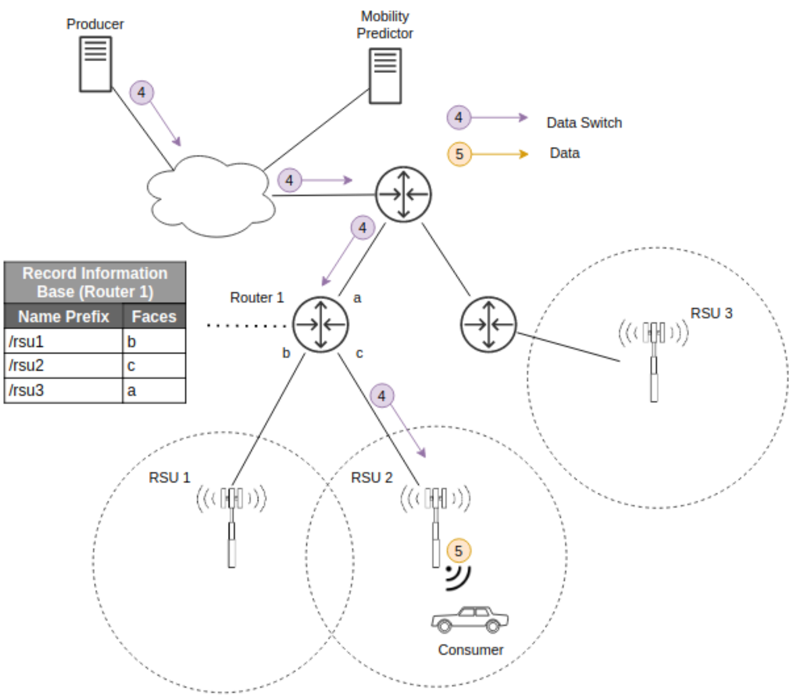


Figure 3.10: Example of transmission of DS packet, and proactive transmission of subsequent *Data* packet.

3.2.2 Pub-sub mobility

When the Consumer opts for a pub-sub NDN based communication, instead of approaching *Single-Data-Request*, it is expected the acquaintance of many desirable contents, with specific Name, through time by only issuing a single *Interest* packet.

Using a pub-sub model resorting to persistent PIT entries raises several issues when the Consumer presents a mobile profile: the path formed by the subscription *Interest* packet, to be used by the *Data* packet, gets outdated too frequently. On top of that, a node's *Data* consumption might get interrupted during wireless handover, causing loss of *Data* packets.

The previous solution proposed in Section 3.2.1 does not fully adapt to this type of communication. However, having covered efficient seamless communication for end-users approaching *Single-Data-Request* communication, we prospect that by using some of the implemented node's mechanisms and interactions, and by introducing new ones, one could achieve seamless pub-sub communication for end-users.

To address these issues, our solution establishes the communications' path, before or during the transition of the mobile node. We preventively solve the path failure by setting the infrastructure to fix the subscription path from the next AP/RSU all the way to the Publisher. When this is done, there is no need to wait for timeouts to trigger the new subscription request.

Forecasting notification

In this solution, the responsibility for detecting potential migrations of mobile nodes between RSUs is also exclusive to the MM, exactly as presented in Section 3.2.1. However, the implemented entity's behaviour for this solution differs. Upon detecting the migration, without the issuing of an *Interest* packet per prediction query, the MM notifies the RSU to which the mobile node is connected. This way, it is avoided the transmission of additional packets sent by the RSUs asking for eventual changes in the topology, reducing the network overhead. Notice that the forecasting is still made with the data of mobile neighbors which is gathered from the RSUs. The same is delivered to the MM in a pub-sub approach, as explained in section 3.2.1. Having such knowledge, the MM is able to monitor in which RSU the mobile device is hosted. Furthermore, in case of potential migration, the forecasting entity knows to which RSU it needs to get instructed.

The main motivation for such approach is the fact that the number of sent *Persistent Interests* is limited. This is intended since the topology is dealing with a pub-sub communication model. It would not be feasible to request for *next RSUs* predictions only at the instant a Consumer reissues its *Persistent Interest* for the pub-sub topic. Specially, it would not behave well if the mentioned reestablishment of the pub-sub communication are set to be at long period of times: When the Consumer is about to migrate to another RSU, no *Interest* is yet programmed to be sent and, therefore, no seamless connection transmission could be fixed.

To handle these Notifications, it is established another pub-sub communication where RSUs subscribe the MM forecasting service. For an RSU to receive these notifications, it needs to send a *Persistent Interest* to the Mobility Predictor. The packet's prefix must be according to the format established by the MM, containing sub-prefixes such as:

- Identification of the MM: the prefix of the Mobility predictor itself (Example: /mm1).
- Action: the type of action or content queried (Example: /notification).
- Identification of the RSU: the name prefix of the respective RSU (Example: /rsu).

An example of a name for the *Persistent Interest* in the current scenario would be:

/mm1/notification/rsu3

The number of dedicated notification topics are equal to the number of RSUs which subscribe to the MM *next RSU* notifications. Once the MM receives a packet of this type, it composes the topic. When it detects potential *next RSUs* for a mobile Consumer, the related information about the occasion is transmitted through notifications. It is dispatched to the topic which corresponds to the RSU hosting the potential emigrant mobile device.

In Figure 3.11 it can be observed that the MM detects potential *next RSUs* for a mobile Consumer hosted by RSU 2. The same Consumer is subscribed to a content topic. Having knowledge about plausible migration, the MM sends a notification to the RSU 3 (packet 1).

Support packets

The support packets introduced for this approach to achieve seamless communication are almost identical to the ones introduced at *Support packets* from section 3.2.1. However, one distinct factor is that these packets include the *LpPacket* tag field *Subscription*, indicating that the established communication is to be of pub-sub approach. To perform these actions, two new types of packets were introduced: Subscription Interest Switch (SIS) and Subscription Data Switch (SDS):

- **SIS**: it is a derivative of the IS packet, that follows the FIB entries of the intermediate nodes, until it reaches the specified Publisher.
- **SDS**: it is a derivative of the DS packet, containing fields specifying the new recipients. It follows both PIT and RIB entries. The consequent processing of the packet will later be detailed in this section, in *Subscription Data Switch Dissemination*.

The SDS has an additional field, apart from the ones introduced at DS, called *PersistentPitEntryLifetime*: it contains the lifetime of the consequent persistent PIT entries associated to the output faces which the packet flows through. This is done so the new pub-sub communication could be established only for the valid duration of the subscription.

In case the *SendBack* field of the SIS is not true, the packet will not update the persistent PIT entry lifetime associated to the *face* of the node it traversed.

Monitoring the Consumer mobility

Algorithm 4 explains the mechanism implemented in the RSUs, so they can handle *Persistent Interests* of its neighbors mobile consumers. This mechanism is important to accomplish seamless connectivity for potential handovers.

When a mobile Consumer sends an *Interest* with the subscription tag (which is depicted in Figure 3.11), the RSU persists the MAC address of the mobile Consumer, along with the name prefix of the content and the correspondent time the subscription expires (line 5). The later is calculated with the Interest's field *Lifetime*. A Consumer can be associated with more than one subscription topic.

Algorithm 4: RSU *Persistent Interest* preprocessing.

```
1  Algorithm processConsumerPersistentInterest(Interest I, String MACConsumer)
2  lifetime = getInterestLifetime(I)
3  name = getName(I)
4  if !consumerHasTopic(name) then
5  |  storeConsumerTopic(MACConsumer, name, lifetime)
6  else
7  |  topicLifetime = getRemainingLifetime(MACConsumer, name)
8  |  if topicLifetime < lifetime then
9  |  |  updateTopicLifetime(MACConsumer, name, lifetime)
10 |  end
11 end
12 nextRsus = getNextRsus(MACConsumer)
13 if nextRsus != null then
14 |  sendBack = containsThisRsu(nextRsus)
15 |  sis = createSubscribeInterestSwitch(interest,nextRsus,sendBack)
16 |  sendInterest(sis)
17 end
```

In Figure 3.11, the message x is the subscribed data by the Consumer, and the message 1 is the notification message that comes from the MM. Upon receiving the notification, the RSU stores the association between the MAC address of the device, potential *next RSUs* and their respective likelihood. This data is obtained from the forecasting entity. When it is noticed the take off of a particular hosted node, the RSU erases all records and associations made between the device, *next RSUs* and respective subscription topics.

Reacting to topology modifications

When the RSU receives a notification from the MM, it proceeds by setting up the tracked subscription connections of the denoted mobile device to the potential *next RSUs*. This process is depicted in Algorithm 5, which represents the processing in the RSUs when a notification is received. If the mobile node is querying a subscription to a new topic, and the RSU is already aware of the possible migrations of the inquirer, then it handles instantly the arrangement of the new subscription connections of the denoted mobile device to its potential *next RSUs* (lines 12-17). In such cases, the new connection would be then established before the conclusion of the node's handover. The process is depicted in Figure 3.12.

The arrangement of new pub-sub communication topics of a mobile Consumer by the RSU is handled with SIS packets (the message 2). It sends SIS per subscribed content of the potential migrating mobile node containing the respective name prefix of the content, the name prefixes of the authorized potential RSUs in fields *RecipientPrefix*, and the remaining *Interes* lifetime of the correspondent subscription topic. Unlike native solutions, the SIS packet

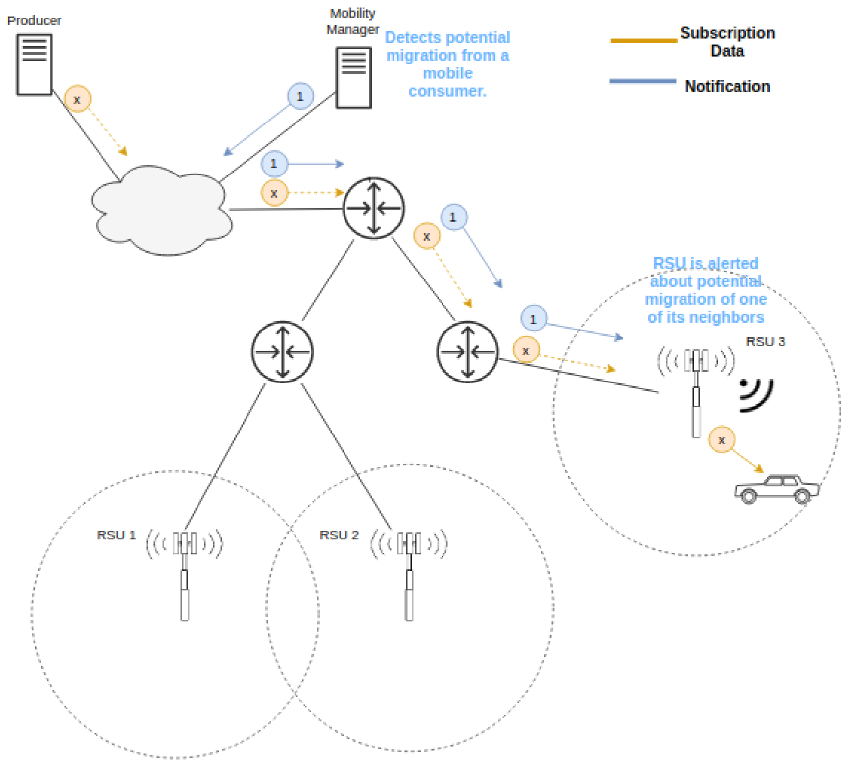


Figure 3.11: Notification transmission to trigger content bifurcation.

follows the path established by FIB entries, whether the intermediate nodes have PIT entries or not.

Algorithm 5: RSU reacting to topology notifications.

```
Algorithm predictionNotification(Data D)
Content = getContent(D)
NextRsus = getNextRsus(Content)
ConsumerMAC = getConsumerMAC(Content)
ApprovedNextRsus = filterNextRsus(NextRsus)
if ApprovedNextRsus != null then
    clearConsumerNextRsus(ConsumerMAC)
    storeConsumerNextRsus(ConsumerMAC, NextRsus)
    subscribeTopics = getSubscribeTopics(ConsumerMAC)
    for topic: subscribeTopics do
        sendBack = containsThisRsu(ApprovedNextRsus)
        lifetime = getTopicRemainingInterestLifetime(ConsumerMAC, topic)
        sis = createSubscribeInterestSwitch(topic,nextRsus,sendBack,lifetime)
        sendInterest(sis)
    end
end
```

The Publisher has to be warned about the new recipients expecting to consume from the topic it provides, which occurs with the reception of an SIS. Thereafter, the Producer creates the SDS packet which would additionally flow through the shortest path to the denoted new recipients, apart from the communication path already established with PIT entries.

Hence, once received an SIS, instead of composing a regular *Data* packet, the Publisher creates an SDS message (the message 3), according to the information provided by the received SIS packets. More precisely, it gets the denoted name prefixes of the new recipients, receives the subscription tag, and it is transmitted to the network. Then, it follows the PIT entries of the intermediate nodes, or it follows the RIB entries for new communication paths to the *next RSUs*.

Reaching the mobile Consumer

The bifurcation process operates the same way as in Section 3.2.1. However, for the new paths pointed exclusively by the RIB entries, it creates persistent PIT entries for these faces. This is done in order for the next topic's *Data* packets to traverse the intermediate node by just following the PIT entries.

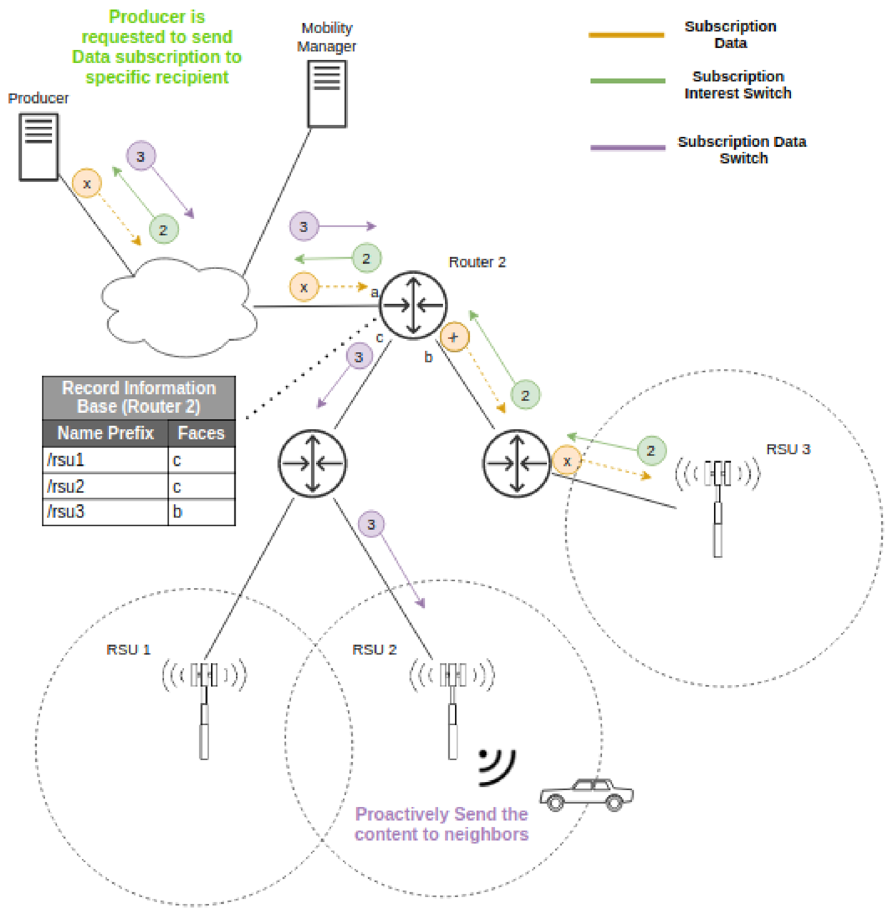


Figure 3.12: Content subscription bifurcation established by a *Subscription Data Switch* packet.

Once an RSU receives an SDS, it checks if its prefix is one of the recipient prefixes in the packet. In that case, it proactively transmits the Data, as a regular *Data* Packet, to the wireless interfaces in order to increase the probability of content delivery to newly arrived mobile Consumers. Additionally, it also creates a persistent PIT entry, associating the content prefix to the wireless interface.

Even after all steps described above have been made, the RSU that hosts a communication defined by the SDS is not yet contextualized about the mobile Consumer subscribed contents, because if the periodic reissue of *Interest* is not settled on the wireless interface, the RSU will not have the necessary information to enable communication after migration at the new RSU. To overcome this, the Consumer reissues the *Interest* packet to acknowledge that he is still interested in the subscribed topic. That being said, if the hostess RSU does not have records about the subscription topics of a Consumer, and the later is about to travel to another RSU's transmission area, then it is not guaranteed that the node would have seamless connectivity during transition.

3.3 Chapter considerations

In this chapter, the conceptual proposal of the ICN architecture to attempt to achieve seamless communication has been described. The chapter started by describing the base work to extend through the proposed solutions. An analysis of the NDN architecture was made, focusing on its mechanisms and structures originally suggested, besides the link protocol used by NDN (NDNLP).

After this review, it was presented the connectivity issues caused by the mobility of the Consumer. This is done for the *Single-Data-Request* and pub-sub NDN communication models. Furthermore, proposals were made to tackle these issues, introducing entities and their mechanisms, such as RSUs and MM, together with their interactions with the topology; presenting supporting packets such as IS, DS, SIS, SDS; and detailing modifications at the forwarding module of the intermediate nodes to process and forward all inquiries and content packets from the proposed solutions.

The solutions presented are dependent of a forecasting module, which is hosted by a remote entity in the topology. That being said, the success of this solution is mostly dependent and influenced by the accuracy of the chosen forecasting application. Furthermore, regarding the forecasting algorithm, it may require temporarily spaced instances about the consumers.

Also, the RSU does not only behave as a bridge for the wireless consumers to connect to the NDN network, but also as a support for MM. The RSU is responsible of contextualizing the MM about its neighbour mobile devices, using a pub-sub approach. It is a crucial module that starts the establishment of seamless connection for the neighbors and their potential RSUs that they are about to handover to.

CHAPTER 4

Implementation

To evaluate the proposed ICN architecture, we used simulation tools to reproduce a virtual infrastructure able to emulate the real ICN. NDNSim is the chosen framework, built from Network Simulator (NS-3), since it provides most features from the NDN communication model.
This chapter covers the NDNSim and its main components, and describes the most relevant changes made to the NDNSim software, to address a better performance in the communication in the face of consumer mobility, considering the proposed solutions presented in this dissertation.

4.1 NDN simulator overview

The NDNSim [54], [55] is an open-source simulator platform, built on top of NS-3 software [56]. The later is a discrete-event network simulator, seen as a network-layer protocol. It operates with several link-layer protocols: point-to-point, Carrier-Sense Multiple Access (CSMA), wireless, among others. Regarding NDNSim, it is widely used by the research community for the developments on the top of NDN and NDN-based architectures. Therefore, to simulate the exchange of NDN traffic among simulated nodes, it also contains functionalities to trace the exchange events. This simulator is on the work to transition to version 2.7, hence it is used a more stable version, which is the previous version, number 2.6.

NDNSim presents several features; the ones that stand out are the following [57]:

- *Interest/Data* format and hierarchical naming scheme;
- Packet processing tables: CS, PIT, FIB
- basic Forwarding Strategy Abstraction.
- Face Abstraction, to support the interface between the NDN and upper (applications) and lower (network, link, transport) layers.

The simulator uses the Named Data Networking Forwarding Daemon (NFD) source code [58]. It is a network forwarder that manages, implements and develops alongside the NDN protocol. This application is present in the main structures: PIT, FIB and CS. The information provided by these structures is used by the forwarding decisions and *face*. The *face*, the module responsible for communication abstraction used by the lower-level transport mechanism, also serves as a bridge between the application layer and the NDN stack.

Most of NDN operations in NFD use *ndn-cxx* [59], a library providing routines for packet encoding and decoding, set of security mechanisms, such as signing and verification, as well as a special application-directed *face* realization.

The integration with NFD and *ndn-cxx* library ensures that simulations are similar to real scenarios, allowing the replication of the development in real environments without significant changes.

For a better understanding of the implemented work, it is briefly presented the NDNSim ecosystem [60], which consists of: *Core NDN Protocol*, *ndnSim Utilities*, *ndnSim helpers*.

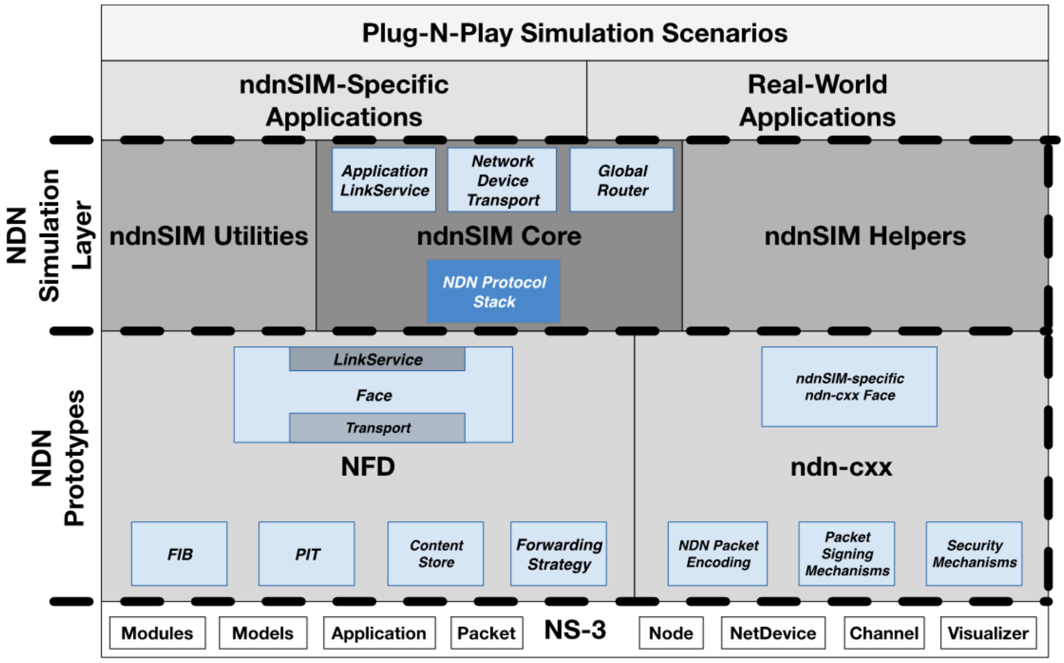


Figure 4.1: NDNSim ecosystem [60].

The core of the NDNSim is accountable for implementing the NDN protocol stack. It is also responsible to communicate with the NS-3 ecosystem by means of *NetDevices* (network layer to device interface). It is the entry point for decoding and analysing the NDN packets, as well as for acquainting information from the structures PIT, FIB and CS, mainly responsible for the forwarding of the packets. Finally, to set up static network topologies, the core of NDNSim includes a global router module for static configuration of FIB routes.

NDNSim possesses a number of packet tracers to obtain simulation results regarding the link, network, and application-level.

Helpers allow a more effortlessly creation, configuration and management for the NFD software components. A set of helpers are also used for the NDN stack and application parameters on NDN nodes.

4.1.1 Drawbacks and challenges

NLSR is not directly integrated with the NDN architecture, neither with the NDNSim, so there is no routing protocol implemented in the simulator. Thus, the RIB loses its functionality, not being able to collect and update the nodes' FIB.

Without a routing protocol, forwarding strategies become inefficient. This also accentuates the fact that NDNSim is not ready for a mobile environment directly. Therefore, the definition of the topology is made at the beginning of the simulation, and its change and dynamism can mean the loss of the known routes.

4.2 Support forwarding mechanism for mobile Consumers

Several changes and additions were made to the NDNSim in order to cover the schemes proposed in Chapter 3.

4.2.1 Wireless interfaces

With the default configuration of the NDNSim, all interfaces were recognized as point-to-point. In an attempt to differentiate both wireless and wired interfaces, both *face* and *LinkService* modules have been changed. A new *face* type was attributed to identify and support wireless interfaces (IEEE 802.11n and IEEE 802.11p). Having realized such modification, a node behavior triggered by a receive message is able to change according to the logic associated with the respective input interface type.

Since the communication between mobile nodes and RSUs are exclusively wireless, the RSUs can take advantage of the aforementioned and differentiate an *Interest* packet incoming from a neighbor mobile node from one that has only traversed the wired infrastructure.

4.2.2 MAC address

Each mobile device is uniquely identified by its MAC address. This makes the use of a device identification allocation protocol needless. The afforded mobility predictor, hosted in the Mobility Manager (MM), requires the identification of each node to accomplish a right prediction, which depends of the respective node's RSU attachment history. Hence, in order for an RSU to populate the database of MM and also to request prediction from a specific device, it has to hand the MAC address of the respective device into its outgoing packets.

In a real scenario, the MAC address could be obtained from the original sender from the Data Link Layer of the *Interest* packet. However, the simulator does not provide a method to acquire the sender's MAC address from a NDNLP packet. To tackle this issue, the mainly required changes and additions are the following:

- **ndn::lp::Mac**: A new NDNLP *tag* was created in order to include the MAC address information. This module is responsible to encode and decode a wire element of NDN-TLV packet format, which is represented as a buffer of unsigned 8-bit integer. A parser operation to String type is included in order for better readability by the RSU and MM. Further modifications were made in: **ndn::Tags**, **ndn::Fields**, **ndn::tlv** for the inclusion and identification of this specific tag, among the others already configured.

- **ndn::Consumer, ndn::ConsumerZipfMandelbrot & ndn::Subscriber**: The consumer application, at startup phase, acquires the automatically designated MAC address of its wireless interface. It is obtained from the node's respective ns3::WifiNetDevice, which holds together all Wifi-related objects. When the delegated consumer application is about to send an *Interest*, it sets the *tag* into the NDNLP packet. The message is then encoded and sent.
- **ndn::Rsu**: Every time an *Interest* is received wirelessly, it checks for the MAC address tags, it parses the unsigned 8-bit integer buffer to String for further decisions: either to communicate to the MM about the respective the neighbor node, or search for notified potential migration warnings associated to the correspondent mobile device. If the consumer requests subscription to a content topic, it also associates the MAC address, the topic's name and the *Persistent Interest* lifetime. This is later described in Section 4.2.7.

4.2.3 Beacons and RSSI

It was proposed in Chapter 3 that the information provided to the MM module would have been derived from the beacons transmitted by the mobile devices. This information would be used to populate the database and to request the intended forecasting. This information is the following: the MAC address of the neighbor wireless interface and the RSSI associated with the beacon. However, NDNSim does not have implemented beacon messages, neither provides a way to determine their associated RSSI values. To tackle this issue, it was decided to replace the RSSI metric with the distance between an RSU and the respective neighbor. This decision was taken since the deterioration of the RSSI is proportional to the distance between the device's wireless interface/antena and AP. The RSU analyses the simulator devices with the possession of the module *ns3::WifiNetDevice* (hold together all Wifi-related objects). Once obtained, it consequently obtains the respective *MobilityModel* pointer. It is an object with the main responsibility to track the evolution of the position with respect to a Cartesian coordinate system. It affords a method to calculate the distance between this object and another *Mobility Model* Object. If the distance between the RSU node and the wireless mobile node is less or equal than the configured RSU's wireless range, then it is considered for the handover process. This information is then calculated with an infrastructure point of view, and not given by packets coming from the neighbors.

4.2.4 Routing Information Base

The RIB allows the aggregation of routing protocol routes, such as NLSR, described in Section 3.1.3. These routes are processed and updated at the nodes' FIB. The RIB includes its own table, associating *Name* prefixes to outgoing *faces* along a denominated metric. Ideally, considering the established metrics, it points to the shortest route. The RIB Manager is responsible for the management and operation of routing protocols, such as NLSR protocol. Nevertheless, NLSR is not directly integrated with the NDN architecture, neither with the NDNSim.

For the work covered by this dissertation, a crucial component is the RIB to have itself completely setup from the possible *Name* prefix announced by all the RSUs, rather than integrating NLSR protocol, which corresponds to the building process of the RIB tables. In the case the RSU topology is not properly configured at each intermediate node's RIB, including the RSUs themselves, the diffusion of the *Data Switch* would not work as intended, therefore the establishment of the RIB is the one needed for the work.

That being said, a new RIB module was developed, with the needed functionalities, along with modules that aim to populate the table's records. The later is utilized in the infrastructure configuration module in order to map the topology of the infrastructure RSU's manually.

The most important additions and changes made are the following:

- **ndn::Rib**: module that represents the RIB table itself. It contains a set of *RibEntries* objects. It provides operations to create, delete and update *RibEntry* objects;
- **ndn::RibEntry**: module responsible to maintain a list of routes associated with a particular name prefix. A RIB Entry contains the *Name* prefix and a list of routes;
- **ndn::RibNextHop**: a route indicates that contents under a certain name prefix may be available via a certain *nexthop*. It contains information, such as: *Name* prefix, nexthop *face Id*, cost, indicating the preference among multiple routes with the same name prefix. The next hop *face* on a route with lower cost is preferred; origin, indicating who is announcing a route. Regarding the origin, since the topology is manually configured it will be denoted as static: it indicates as a static route;
- **ndn::RibManager**: this module becomes part of the NFD management protocol. It provides commands to register and unregister routes. RIB Management commands and datasets are available under the namespace *ndn:/localhost/nfd/rib*;
- **ndn::RibHelper**: this module interacts with the RIB manager of NFD by sending special *Interest* commands to the manager in order to add or remove a next hop from the RIB table entries or add routes to the RIB manually.

Command Interests are mechanisms for issuing authenticated control commands. Signed commands are in terms of a command Interest's name; these commands are defined to have five additional components after the management name space: command name, timestamp, random-value, *SignatureInfo*, and *SignatureValue*.

4.2.5 Mobility Manager

The MM has its own NDNSim module. However, it is dependent of an external service written in Python.

The NDNSim's application class is in charge of handling all received *Interest* and *Data* messages. It is responsible to parse all given information which may regard to the forecasting algorithm, or storage of data related to the mobile device's infrastructure.

The data extracted from these messages is shared with the Python module. This is responsible for the actual operations of data storage and calculation of the future RSUs of the mobile consumer. The reason for this separation is because Python has a significant number of powerful and easy to use external libraries for data analysis, prepossessing, as well as for implementation of forecasting algorithms (for example: pandas and numpy).

The interchange of information between these two modules is set using sockets. The NDNSim module acts as a client and the other as a server. The interchanged messages have a predefined structure in order to perceive them as commands. It consists of a numeric identification of the command, following the information required regarding the operation. The interactions between these two detached modules and their respective structure message is represented by the sequence diagram illustrated in Figure 4.2. The two most relevant commands are:

- **Mobile Consumer Status Report**: this message comes from the NDNSim module to the Python component. It contains sequential values about the RSU and respective neighbors address and distance with the closest AP. This information is provided by the messages the MM subscribes to each topology's RSU. Therefore, the gathered data is done at the NDNSim MM object, having on top a filtering process which only considers the nearest RSU of a mobile device, thus avoiding complexity. The data is used to periodically contextualize the entity about the status of the infrastructure's mobile devices, by means of storing the data which could be used by the mobility prediction algorithms. Additionally, the MM is set to act upon these messages, such as returning a notification to specific RSUs about potential migrations of its neighbor devices, in case the RSU allows it with subscription request;
- **Forecasting Request**: this message represents the request of the mobile prediction of a specific consumer in the next RSU. Apart from the command identification, the MM NDNSim module forwards the address of the consumer. The python MM component returns a sequence of *next RSUs* names regarding the consumer of choice.

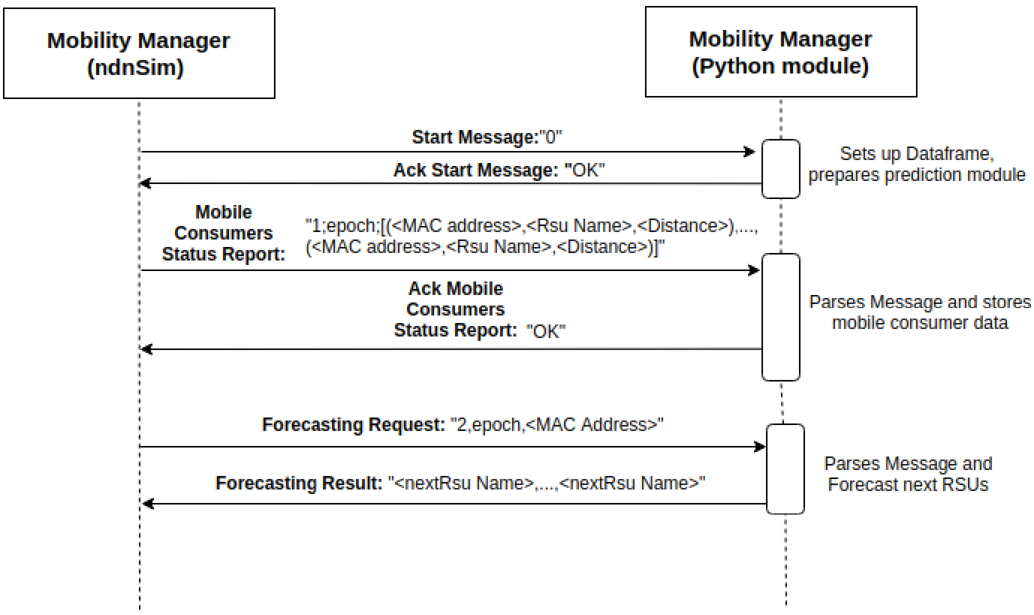


Figure 4.2: Mobility Manager NDNSim and python modules' interaction.

At configuration, it is given the name prefixes of the RSUs that the entity has to subscribe at startup. A timer is set in the case of unsatisfied subscriptions. It is recognized as unsatisfied for a period of time that no content regarding the topic is received. In such cases the MM triggers retransmission of the Persistent *Interest* in concern. The lifetime *Interest* of the subscription is also constant and configurable. Upon expiration, another subscription requests are settle to each infrastructure's RSUs. Each RSU has its own unique topic, as mentioned in Section 3.2. The content involved at these subscriptions is information regarding the state of infrastructure mobile's devices, used as context aware to the module. The *Mobile Consumer Status Report* messages are supported by these packets.

4.2.6 Support packets

Six new types of packet structures are interpreted at the NDNSim: *Persistent Interest*, *Persistent Data*, IS, DS, SIS, SDS.

The *Persistent Interest* and *Persistent Data* consist of a regular *Interest* and *Data* object, respectively, but with a *Subscription* tag. These implies slight differences imposed at the processing and forwarding mechanism of the intermediate node.

For the composition of IS and DS in NDNSim, two classes were made, derivative from the *Interest* and *Data* classes, respectively. Because SDS has an additional field from DS, a dedicated class was also made in order to include the field *PitEntryLifetime*. Moreover, it composes the subscription tag. Regarding the SIS, the simulator interprets it as an IS with a subscription tag (no new class is needed to be made, since both packet type structures share the same fields).

The additional fields for each support packet were defined for each class, along with the definition of the corresponding type-length values. Furthermore, operations of Encoding (*wireEncode* method) and Decoding (*wireDecode*) were overwritten from their predecessor classes in order to compose and parse the NDNLPv2 protocol packets, respectively.

In order to include these operations of encoding and decoding and furthermore enabling processing and forwarding of these new packet objects, modifications to the modules regarding the *face* and *Link Service* were needed: *nfd::face::GenericLinkService*, *nfd::face::LinkService*, *nfd::face::AppLinkService*. These generally represents the Link Service that implements the NDNLPv2 protocol, explained in Section 3.1.1. They are responsible to transmit and receive all types of packet objects, including parsing and packaging operations.

4.2.7 Road Side Unit

The NDNSim possesses a class that all NDN applications, such as *Consumer* and *Producer*, should be derived from, which is *ndn::App*. It allows them to implement virtual calls such as *onInterest* and *onData*. These enables the node's application to add another processing layer to the packet, allowing to make decisions from the given packets, regarding for instance forwarding and caching methodologies.

Therefore, it was created a class ndn::Rsu, concerning the RSU application, derivative from the application class, since it is supposed to react to messages received from entities such as Consumers, CP and MM, following the behaviour described in Chapter 3.

RSU as a content provider

An RSU provides information concerning its neighbor mobile nodes, their MAC address and their respective measured distances under the coverage area. The retrieval process of this information is mentioned in Section 4.2.3. This content has a name, which also uniquely designates the RSU. The name is set at the configuration of the RSU. Its topology is known at each intermediate nodes' RIB table, and thus the FIB itself. When an *Interest* reaches the RSU (with the same name that designates the RSU) asking for information of its neighbors, it returns a *Data* packet containing the description of its neighbors. If the same *Interest* has a subscription tag, then it sends the same *Data* earlier mentioned but with the subscription tag (used to set the subscription path towards the inquirer position). This tag will trigger a timer which sends periodically *Persistent Data* containing the content in concern. The data rate for this content is instantiated at the RSU configuration as well.

The MM uses subscription to obtain the aforementioned data from the Infrastructure RSUs, alleviating the network's traffic from the corresponding regular *Interest* packets.

The node hosting the RSU application contains a CS. Therefore, without the implication of the RSU object, if the content requested by a neighbor is cached at the mentioned structure, it directly replies, through the incoming *face* of the *Interest*, with a correspondent *Data* packet.

RSU as inquirer of mobile prediction

The knowledge about future RSUs of the hosted neighbor mobile devices is the key element for the scheme covered by this dissertation to work properly. These are predicted by the MM, and the information that it regards is made of use by the RSUs.

There are two ways for an RSU to obtain such information from the MM, has mentioned in Sections 3.2.1 and 3.2.2: by a single request of data using the *Interest* packet; or through subscription of notifications regarding potential migration of its neighbor nodes. One of these modes is set at the configuration of RSUs.

In case the second mode is chosen, the data about the potential future RSUs of specific neighbors is stored, mapping the address of the neighbor with a set of names corresponding to the *next RSUs*. The association is replaced if another notification regarding the same neighbor is received. The mapping is deleted if the RSU no longer detects the mobile device inside the coverage area. This neighbor awareness is done periodically and its periodicity is set at the configuration.

RSU as a service for seamless connectivity

As denoted in Chapter 3, the RSU does not only work as a bridge for mobile Consumers to communicate with ISP, but also surges as an entity responsible to manage the establishment of communication between the content provider and the future RSU of the neighbor mobile node for seamless handover. For a better understanding about how the RSU serves the consumer, it is illustrated in Figure 4.3 the involvement of the different established state mappings regarding the neighbors according to some of the packets the RSU application received.

The consumer is set to transmit the *Interest* with a tag specifying its MAC address. In case the RSU captures, at the wireless interface, an *Interest* without the MAC tag, it routes the packet as a regular intermediate node does. Otherwise, it goes the routing through another processing layer.

In case the Prediction Subscription mode is set on a RSU, when a regular *Interest* is received, it checks the established mapping between the MAC address and the set of *next RSUs*. If the association is checked, the RSU composes an IS based on the *Interest* and the mapped *next RSUs*. Otherwise, it sends an *Interest* packet. This sequence is represented by the green arrows from Figure 4.3.

If the aforementioned mode is unsettled, the RSU proceeds by requesting mobility prediction regarding the mobility predictor designated by the given MAC address included in the received *Interest* packet. As mentioned in Section 4.3, this request is also done in the format of an *Interest* packet. The RSU creates a sequence number, mapping the association of this number with the received *Interest*. Moreover, the same sequence number is included as a component in the name of the *Interest* regarding the prediction. A configured timeout is set, in case no response is given from the MM, proceeding by the transmission of the regular *Interest*, and deletion of the association with the sequence number. Once a data packet proceeding from the MM is received (the RSU recognizes its source by the structure of the name), the RSU extracts the sequence number in order to get the stored *Interest*. Having both *Interest* and *next RSU* information, the node proceeds with the creation of IS.

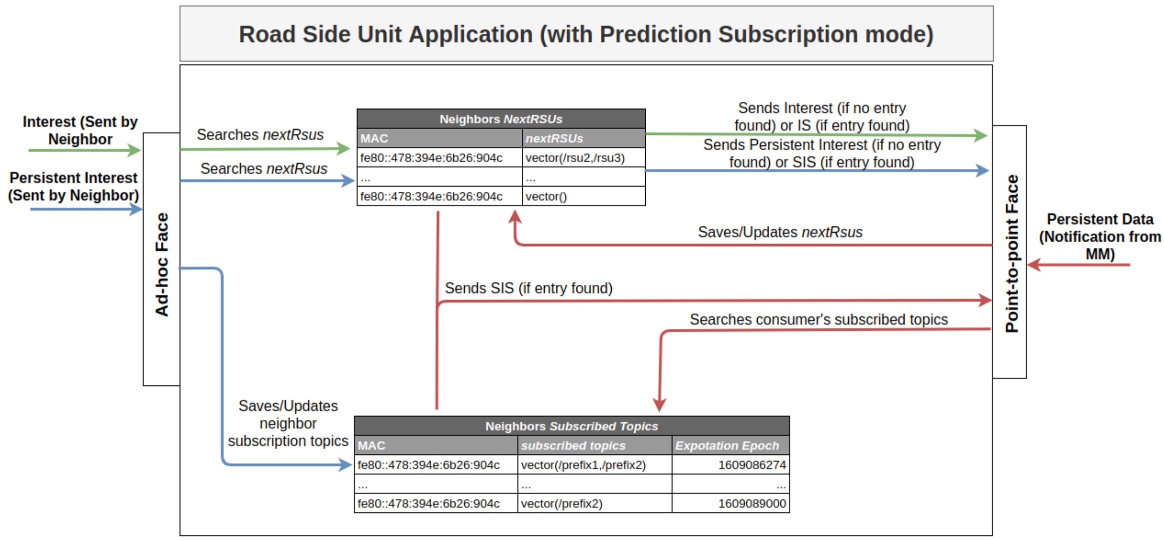


Figure 4.3: RSU Application.

In case a *Persistent Interest* is received at the wireless interface, the RSU establishes another association: with a map class object, it is stored the association of consumer MAC address of the neighbor consumer, the name of the content it is subscribing to, and the time of expiry. These associations are deleted after the *Interest* lifetime expiration or after the node is no longer detected by the RSU. If *next RSUs* regarding the consumer are known, then an SIS is composed. The set *Interest* lifetime defined in the packet is equal to the remaining time until expiration of the content subscription. The mentioned operation is represented by the sequence of blue arrows in Figure 4.3.

Additionally, when a notification is received from the MM, it checks the records from both *Neighbor Subscribed Topics* and *Neighbors next RSUs*. When both are checked, acknowledging the respective consumer content subscription and potential future RSUs, then the RSU proceeds by composing the SIS. The set *Interest* lifetime defined in the packet is equal to the remaining time until expiration of the content subscription. This process is already mentioned in Section 3.2.2. Having this in consideration, if the Prediction Subscription mode is not enabled, seamless handover for pub-sub communication is not handled.

RSU as a proactive content provider

Upon reception of a DS or SDS, the RSU checks among the specified *next RSUs* if the name which the RSU is designated is included. If it is the case, the content of the packet is stored at the node's CS using the configured cache policy, following with a proactive transmission of the content to the ad-hoc *faces* of the node with the format of a *Data* (in case of a a DS) packet or *Persistent Data* (in case of a SDS). If the later is transmitted, a persistent PIT entry is set regarding the content and selected *face*. The *Interest* lifetime is equal to the one denoted at the field *PitInterestLifetime* from the SDS.

4.2.8 Forwarding

The Forwarder class comes as the mediator of all the processes from the NDN protocol stack, being the core point to gather and process information over the course of the forwarding procedure. The implemented system is changed to comprise the proposed modifications introduced in Chapter 3.

PIT

Apart from the addition of another table (RIB), modifications at the existing PIT were also made. In order to understand the additions and modifications, this section will briefly introduce classes that compose the PIT from the NDNSim. This section also presents an Unified Modeling Language (UML) class diagram in Figure 4.4 for a better understanding:

- ***nfd::Pit***: it represents the *Interest* Table itself. It is responsible for the state of the PIT table of the node. It contains a vector of *nfd::PitEntry* objects. Moreover, it contains the search, read, insertion, and deletion operations of objects of class *nfd::PitEntry*;
- ***nfd::pit::Entry***: it represents an Entry of the PIT. It is responsible to make association between a name of the incoming *Interest* name and all incoming and outgoing node's *faces* of the packets. This mapping is established with a vector of *nfd::pit::InRecord* and *nfd::pit::OutRecord* objects. Operations concerning read, deletion, insertion and search are included;
- ***nfd::pit::InRecord***: contains information about an *Interest* from an incoming face;
- ***nfd::pit::OutRecord***: contains information about an *Interest* from an outgoing face.

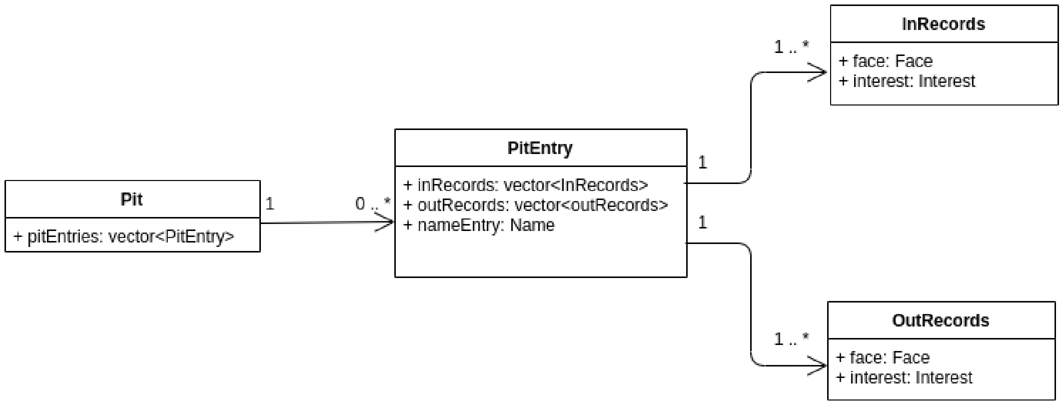


Figure 4.4: UML class diagram representing the NDNSim PIT.

Persistent Interest & Persistent Data Forwarding

To simulate pub-sub communications using *Persistent* PIT entries, such as the approaches mentioned in Section 3.1.2, we resort to the *LpPacket* tags. A new tag, named *Subscription*, is composed in order to arrange the NDN packets so they can set up or get going a *Subscription* path, from the *Publisher* to the *Subscriber*.

A *Persistent Interest* consists of an *Interest* with *Subscription* Tag. The *Forwarding* process applied to these packets is similar to the one established for a regular *Interest*. The *Persistent Interest* is dropped when the active PIT entry's lifetime (information retrieved from an *InRecord* object) is higher to the lifetime specified at the *Persistent Interest*. Otherwise, the packet reaches the CP. Both situations set/update the pub-sub communication path between the *Subscriber* and the *Publisher* until expiration. A *Persistent Data* consists of a *Data* with a *Subscription Tag*.

When the Publisher receives a *Persistent Interest*, a *Persistent Data* is composed and sent out. The Forwarding process applied is similar to the one established for regular *Data*. However, the introduced one does not remove the respective node's PIT entries (the InRecords objects as well), and neither caches the content in the CS. Hence, the pub-sub communication path is maintained and only recent content yield by the CP is received by the *Subscriber*.

IS & SIS Forwarding

The forwarding processes of IS and SIS are very similar to the one already implemented for the *Interest* packets. However, there are additional operations that the packets have to go through, in which the PIT is employed.

The general forwarding process is explained in Section 3.2.1 (*Interest Forwarding*). In order to avoid redundancy of transmission of identical IS (while it has not yet been satisfied), such as it is implemented for *Interest* packets, an additional check has to be made apart from the content name and outgoing face: verification about the *next RSUs* names of the IS is needed in order to drop it if a yet unsatisfied PIT entry already has indicated those names.

In order to establish this filtering mechanism at the forwarding module, modifications were made in modules regarding PIT and associated classes.

Since additional association of *next RSUs* names and associated outgoing faces to the content name is needed, *nfd::pit::OutRecord* was modified. It was included a vector of names corresponding to all *next RSUs* related to the *OutRecord*, and therefore, to the PIT *Entry*.

Moreover, the class *InRecord* was modified in order to have a flag denoting if the *face* should or should not be ignored. It is set to be ignored if the *sendBack* elects it to be. Otherwise, and even with the reception of a regular *Interest*, the PIT entry would not be ignored when the respective *Data* arrives.

Read and write operations regarding the features added were developed as well, for both *InRecord* and *OutRecord* classes. Hence, the forwarding module is now capable to maintain the state of the node's PIT concerning the *Interest Switch* and its additional fields, preventing the transmission of identical packets.

However, this filtering operation is not considered for SIS. This is done because the conclusion of PIT entries does not happen with the reception of a respective single data, but rather with the expiration of a specific lifetime (*Persistent Interest* lifetime). Since the node does not have context about the other nodes persistent PIT entries (which builds pub-sub communication path between CP and new recipient) and their respective lifetime, it would not be appropriate to drop a SIS because an identical one has been dealt before. For example, consider for a specific content topic, that an RSU had already established a pub-sub communication between the CP and a mobile device's next RSU. Consider also that there is another consumer connected to the same AP and it is consuming from the same topic, however at a much superior lifetime subscription. When potential migration is also detected for the same *next RSU*, the persistent PIT entries of the intermediate nodes, which the path between the CP and *next* RSU covers, needs to be updated according to the greater lifetime, in order for the communication not to get interrupted due to lifetime expiration of PIT entries regarding the other consumer. This reason is enough to consider avoiding filtering a consequent drop-off of identical SIS packets. Furthermore, no *next RSUs* is stored in the *OutRecords* when dealing with SIS.

DS and SDS forwarding

Apart from reading operations at the RIB table, the new support data packets (DS and SDS) also implies writing and reading operations at the PIT entries in order to accomplish content transmission bifurcation to reach new denoted recipients by the packet themselves. An overview of the bifurcation in concern is described already in *Data Switch Dissemination*, from Section 3.2.1.

The class *InRecord* was modified in order to state names of the recipients which concerns the node's output faces which the DS and SDS are to be sent.

With the reception of a DS (or SDS), the Forwarding module (*nfd::Forwarder*) searches, at the RIB, for the faces pointed out by the names of *next RSUs*. These are grouped according to the respective outgoing face. Once established, it creates or updates *inRecord* objects regarding the name of the content and *face*, more specifically the vector of recipients name. It is the inserted all related recipient names. Hence, one could say that RIB not only comes as a mean to update the FIB table, but also the PIT in order to transmit content to new recipients. Once the PIT entry is created/updated, the forwarding of *Data* and DS packets is done. It iterates for each *face* where the content in concern has to be sent. If the *InRecord* has *next RSUs* values, it proceeds by adapting the received content packet into a DS containing as *next RSUs* the ones denoted by the *inRecord*. If the *inRecord* does not contain *next RSUs* but *sendBack* flag is enabled, the node adapts the DS received into a regular *Data* packet, sending it to the *face* in regard. If the type in the DS packet does not have the subscription tag, then it deletes the PIT entries after transmission. Otherwise, if it is of type SDS, the PIT entries are updated, removing the *next RSUs* but keeping the PIT entries, so the pub-sub communication persists until the lifetime.

4.3 Chapter considerations

Most of this chapter reflects on the relevant work implemented in order to obtain a simulation tool with the proposed solutions that address seamless consumer mobility in an NDN-based architecture.

A brief description of NDNSim was presented along its key features, covering its different layers, divided by: NS-3, NFD, NDNSim core, and *ndn-cxx* library that supports several layers.

This chapter also covers some of the most significant obstacles at the chosen network simulator, while issuing opted methodologies to tackle these challenges. Finally, the required changes to existing modules and addition of classes were introduced, and it was further explained how the implementation enables the solution to fit in the simulation environment.

CHAPTER 5

Evaluation

Network simulators allow the analysis of different network architectures in a real-world context. The implementation and evaluation of the proposed work was developed using the network simulator ndnSIM. It allowed us to implement and analyse the proposed solutions, explained in Chapter 3, using different topologies and scenarios. Two different network topologies are described, both of them used to evaluate the proposed communication approaches handling the Consumer mobility: Single-Data-Request and pub-sub.
In this chapter it is explained the topology and all the preprocessing work involved to set the scenario up for evaluation. Furthermore, the chapter details the parameters of each scenario and the evaluation metrics. For each scenario, the simulation conditions are described and the results are presented and discussed.

5.1 Evaluation scenarios

This chapter evaluates the efficiency of the solutions presented in Chapter 3, for seamless connectivity between mobile Consumers and content hostess nodes, for both *Single-Data-Request* and pub-sub communication types.

Two different network topologies are built and used for analysis. The first one is used for *Functional* scenarios: these are used to confirm that the functionality of the solutions works as expected. Moreover, mobility traces of the used mobile nodes are fabricated, allowing to freely evaluate scenarios using different number of nodes, different trajectories and nodes' speed range. The other one is influenced by real mobility traces of vehicles: these scenarios are simulated and evaluated to verify the application performance and feasibility in a real context environment.

5.1.1 Functional scenarios

Functional scenarios are used to verify the functionality of an application. They are executed and analysed to validate the mechanisms implemented at each node and application of the topology. They are also aimed to prove, in many different circumstances, that the solution tackles the issues introduced in Sections 3.2.1 and 3.2.2, allowing for consistent connectivity between the mobile Consumer and the network. They also help to understand the implications of having such solutions implemented in the network, for instance in the overhead.

Regarding the Consumers at each scenario, their number can be variable, as well as their behavior. Since the mobility traces are manufactured, we are able to take into account other characteristics applied to the simulation which are not related to the elements of the wired topology. Their trajectory and velocity are free to be modified in the simulation for each scenario. These feature modifications are not applied to real mobility traces, since it would remove the novelty of the scenarios' purpose: to verify the application performance and feasibility in a real context environment.

The topology used for the functional scenarios consists of 36 RSUs and 14 intermediate nodes. Moreover, three additional nodes are wired up to the network, behaving as Producers of content with different prefixes. Also, one MM is connected to the network, through point to point link. A visual representation of the mentioned nodes and their wired connections is illustrated in Figure 5.1.

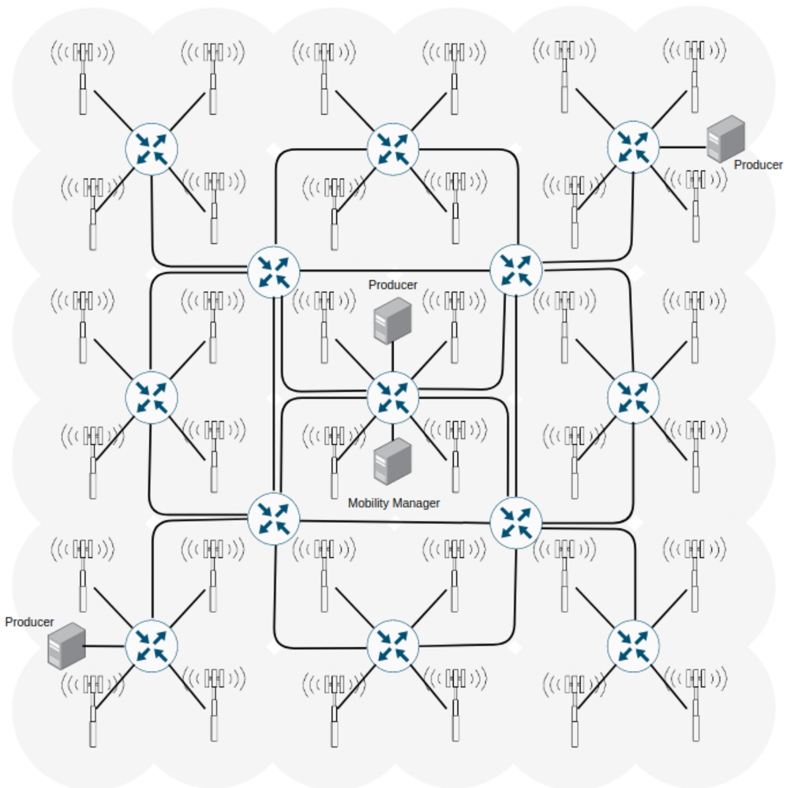


Figure 5.1: Wired topology used for the functional scenarios.

5.1.2 Scenarios using real vehicle mobility traces

To bring our evaluations one step closer to a real scenario, we use two real traces of urban mobility, both from the city of Oporto, Portugal: mobility traces of *Sociedade de Transportes Colectivos do Porto* buses, collected through the Porto Living lab IoT platform [61], and mobility traces of city taxis taken from the Taxi Service Trajectory Prediction Challenge 2015 [1]. In total we consider 25 vehicles (a mix between taxis and buses), 50 RSUs and 19 intermediate nodes, spread throughout the city center as illustrated in Figure 5.2, spanning over a period of four hours.

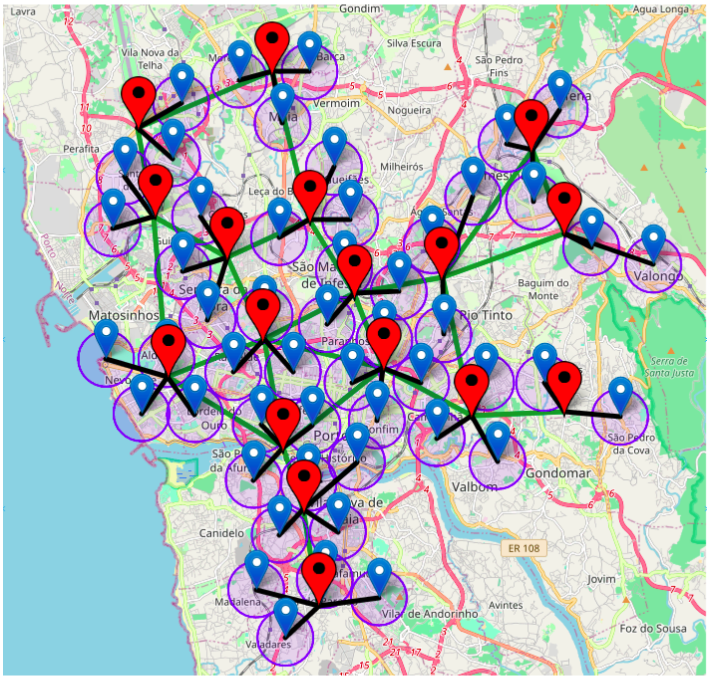


Figure 5.2: Network topology of the evaluated scenario using the city of Porto, Portugal. Backend Routers are represented by the red pins, and RSUs are represented by the blue pins.

Both datasets are from different time periods: trace mobility corresponding to Taxis are from 2018-01-23 00:00:00, Tuesday, to 2018-01-29 23:59:56, Monday; trace mobility corresponding to Buses are from 2013-07-01 01:11:22, Monday, to 2013-07-09 22:05:38, Tuesday. The buses' and taxis' mobility traces were selected and formatted so they could be processed by the *ndnSim*. Moreover, a time adjustment is established, so the traces of the nodes correspond to the same day of the week and hour. A 4-hour window is considered, having in consideration the traffic peak times at a Friday. The mentioned simulation time and the number of Consumers are considered, since the required time for the simulation conclusion is sustainable for a proper analysis of the proposed solutions.

The nodes have been selected with extensive mobility throughout the regions covered by the topology. The greater distances the node travels, the more transitions to other RSUs it has. Hence, it allows to better evaluate our solutions, considering that it resolves connection interruptions due to handovers. This selection has been established by analysing the traces of nodes resorting to heatmaps. This process was established until we got a balanced number of taxis and buses. Considering the selected mobile nodes, the average number of handovers is equal to 11.28 per vehicle, and the total number of handovers is 282.

[1] http://www.geolink.pt/ecmlpkdd2015-challenge/dataset.html

A network topology is built, taking into account the selected real vehicular mobility traces. Once more, heatmaps illustrating the mobility traces and their density were used to select the appropriate positions of the RSUs and their wireless range striving to afford communication coverage throughout most of the Consumers trajectories. The consequent topology used for this type of scenarios is illustrated at Figure 5.2. Considering the RSUs' locations, there are still certain areas with no coverage by the RSUs.

Preprocessing of the real vehicular mobility traces

The original tracing samples of the vehicles are spaced by a 15 seconds window. This is not granular enough for the Consumers to have a smooth transition from one transmission area to another, compromising the results: using a tracing window of 15 seconds, a node will stay at the same position, consuming content at the current RSU where it is located. The mobility from this vehicle is hidden throughout the mentioned time. Moreover, the unseen tracing could represent a handover. Thus, a more strict granularity is important to achieve veracity in the obtained results. This granularity issue is depicted in Figure 5.3.

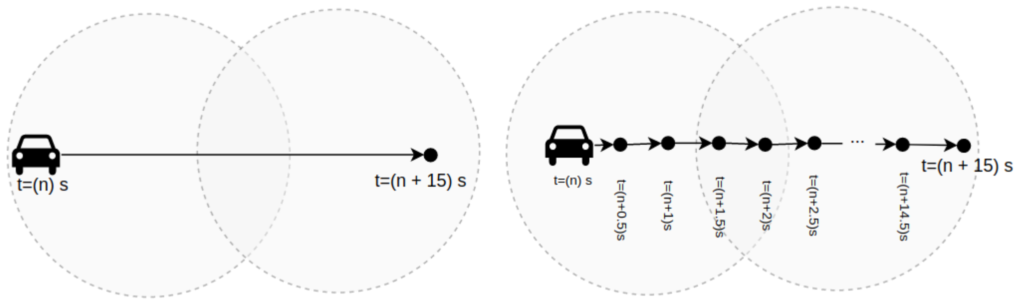


Figure 5.3: Example of non-granular tracing points (left) and a more detailed tracing points of the trajectory of a node (right).

Moreover, the dataset has traces that were considered abnormal, for instance: some records considered a vehicle to travel kilometers away from its last position, in less than 15 seconds. This issue had to be dealt, since it compromises the truthfulness of the mobility trace of the vehicles. The anomalous positions were filtered if the position in consideration had more than half of a kilometer away from the last validated position. Furthermore, to achieve granularity for the vehicles' tracing, new positions for the nodes were established at the linear trajectory, which corresponds to the one seized with a 15 seconds window. Between each trace, the used timestamp was of half of a second. An illustrative example about the fabricated traces, derivative from the real ones, is represented in the right illustration of Figure 5.3. The velocity represented by these new traces, belonging to the original trajectory from the 15 seconds window, is constant, so the distances between the new positions and their adjacent positions are equal.

5.1.3 Mobility Manager

An MM is built for each scenario. The application hosted by this node is aimed to have knowledge about every RSU transaction of the respective infrastructure's mobile Consumers beforehand. In other words, the MM is configured to know exactly the Consumers RSUs transitions and their respective epoch. Every time this node is asked for the prediction of the *next RSUs* for a specific Consumer, it replies by sending the different RSUs the Consumer will have coverage from, in a time ahead. Hence, a time window needs to be configured, so the MM can consult which is the mobile node's *next RSU*. All this information is calculated before the simulation starts.

5.1.4 Simulation pipeline

The setup of these scenarios goes through a process, represented in Figure 5.4, as a pipeline. For each scenario the process undergoes the following tasks:

- *Trace File Generator:* Module in charge of manufacturing the mobility trace of the mobile Consumers. It consists in creating a dataframe which format is compatible with *ndnSim*. The fabricated sequential trajectories are always established according to a preconfigured area of interest. The subsequent destination of a trajectory is chosen randomly, taking into account the established dominion. The number of nodes and their speed range are considered, and they are defined at configuration. This task allows to freely manufacture mobility traces, and therefore build scenarios responsible to evaluate the proposed solutions regarding: the number of Consumers and the velocity range of mobile devices. The granularity of the traces is equal to 0.5 seconds; however, the task is programmed to handle other imposed time granularity [2].
- *Bootstrap Generator:* The records about the mobile Consumers associated RSUs throughout time are built and gathered into a bootstrap, using simulation with *ndnSim*. The topology has the MM node configured so it subscribes, throughout the entire simulation, to the RSUs. This subscription has been explained in Section 3.2.1. The updates are arranged each second, giving the application insight about the mobile nodes (the RSU associated to the mobile node at a given time). Hence, it enables the MM to record the entire history of specific RSU connectivity of each mobile device. It is only considered the nearest RSU which a mobile node is in contact with (inside the wireless range of the AP).
- *Mobility Manager Dataframe:* The aforementioned bootstrap is preprocessed and simplified. For each mobile node, it only considers the RSU and the instant it entered to its transmission area. The resulted dataframe is consulted by the MM at simulation phase in order to search for the future RSUs of a specific mobile node, regarding the current simulation time, when prediction is requested.
- *Simulation*: Once the mobility traces and MM are settled, the scenario is then ready for simulation.

[2] For simulations regarding scenarios using real mobility traces, the first task of the pipeline (*Trace File Generator*) is ignored.

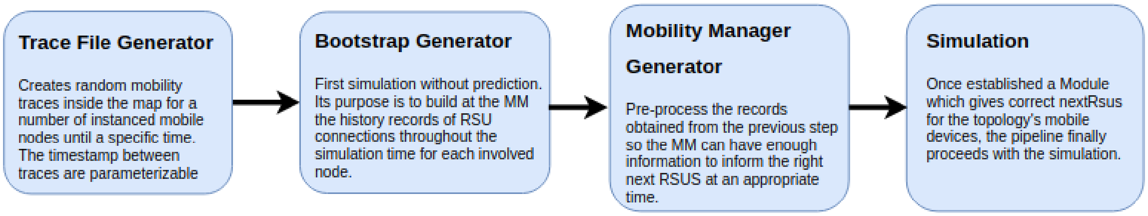


Figure 5.4: Pipeline to setup network simulation and MM using the proposed solution for Consumer Mobility.

5.1.5 Parameterization

Both network topologies are imposed to several circumstances. The purpose is to understand the impact of the developed solutions at vasts conditions which the topology is imposed to. These are configured as environment variables either at the Trace File Generator or at *ndnSIM* simulation run. The parameters that will vary in each simulation are the following:

- *Wireless range*: Communication coverage of the topology RSUs and mobile devices. The wireless communication set for the RSUs is not large enough to cover other RSUs, in order to avoid unnecessary communication between them. The variable wireless range helps to analyse the behaviour of the solutions from smooth (large coverage) and non smooth (small coverage, increasing the chance of having no coverage) handovers of the mobile nodes.
- *Interest frequency*: This variable is important in scenarios which evaluate the solution for the *Single-Data-Request* communication type. Each scenario will have the Consumer requesting for specific contents at the same rate as the others. It is aimed to analyse how the solution behaves from non-congested network scenarios to saturated environments.
- *Number of Consumers*: This enables different instantiation of the number of Consumers.
- *Consumers' speed range*: This is only important in the Functional Scenarios. It is expected that the faster the node is to transition to another RSU's transmission area, the more likely its communication is to get interrupted. We aim to verify this hypothesis, as well as to validate the efficiency of the developed solutions.
- *Producer delay*: When a Producer does not have a requested content cached, it has to fabricate the solicited *Data*. Thus, there is a processing time involved at the entity caused by the creation of the *Data* packet. The greater this time is, the later the packet will arrive at the correspondent RSU. This might increase the chance for the petitioner to be unable to retrieve the content because of a sudden handover. It is of interest to validate this and understand if the solution is suitable for different Producer delay values. This will only be used for the Scenarios which are to evaluate the *Single-Data-Request* Consumer mobility solution.
- *Link delay*: The Round Trip Time (RTT) is also partially caused by the delay of the data-link. Consequences of having large link delays are equivalent from the one introduced at the Producer delay.
- *Persistent Interest* lifetime: This is important in the scenarios that evaluate pub-sub Consumer mobility. Each Consumer is set to reissue the *Persistent Interest* to the

topic in interest when the PIT entry is expired. At the occasions where a Consumer migrates to another RSU transmission area, the longer the expiration is, the longer is the reestablishment of the communication in consideration using pub-sub native solutions. As mentioned in Chapter 4, the main motive for the pub-sub solution is to avoid this packet loss before the subscription renewal.

5.1.6 Metric evaluation

Two main types of metrics are considered to get the performance of the proposed forwarding solutions: *Satisfied Interests* and *Network Overhead*. The first metric measures the Consumer effectiveness; the second one assesses the efficiency from a network standpoint. Specific aspects of these are given below:

Satisfaction ratio

The added support for node mobility, both from a *Single-Data-Request* and pub-sub approach, allows the infrastructure to react directly to changes in path availability and, therefore, properly adjust how packets are forwarded towards their destination.

We study the number of *Interests* arrived at their corresponding Publishers/Producers and how many of the *Data* packets, sent upon receipt of the *Interest*, actually arrived at the Consumer/Subscriber. This helps to understand if the adjustments to node mobility reflect in a positive outcome. In the case of the solution for pub-sub communication, by allowing the Subscriber node to inform the Publisher node on the rate at which data should be transmitted, we can calculate the expected number of *Data* to be received by the Subscriber.

It is also considered the metric Timeout *Interests* corresponding to the number of *Interest* packets expired without receiving a *Data* reply, regarding the Consumers.

Network overhead

It is important to study the overhead caused by the proposed solutions. It is also important to notice that overhead escalates in an ad-hoc network and can lead to congestion, impacting directly the network performance. The following metrics were considered:

- *OutInterests* & *OutInterestsSwitch*: number of *Interests* sent by the nodes in a global network scale.
- *OutData* & *OutDataSwitch*: *Data* packets equivalent to the *OutInterests* metrics.

5.2 Evaluation

5.2.1 *Single-Data-Request* communication model

Functional scenarios

This section analyses the proposed *Single-Data-Request* solution covering Consumer mobility awareness, from now one simply denoted as SDR-Mob, which is compared with the NDN native strategy. After a general analysis between the two solutions, we study how SDR-Mob is impacted with different number of mobile nodes, RSUs and mobile nodes' wireless range, *Interest* frequency, mobile nodes' speed range, link delay and Producer delay. The MM responsible for forecasting the node's *next RSUs* is obtained from the pipeline mentioned in Section 5.1.4. It is configured with a 3 seconds window (equivalent to the Consumer *Interest* Timeout), *i.e*, when asked for prediction, it looks further in time (3 seconds ahead) and checks from the elaborated dataframe set in the *Mobility Manager Generator* if the respective mobile Consumer will be in a different RSU coverage area. The remaining configurations are illustrated in Table 5.1. Notice that the results presented in this section are not cumulative, but per Consumer, and therefore, they contain the mean and 95% confidence intervals.

Table 5.1: Baseline simulation configurations - *Single-Data-Request* functional scenario.

Average velocity	50 $\pm 10 units/s$
Number of mobile Consumers	25 nodes
Delay link	120 ms
Number of existing contents per prefix	70
Frequency	5 packets/s
Wireless range	150 units
Producer delay	500 ms
Content Store size	30
Consumer timeout	3 seconds
Interest lifetime	3 seconds

Figure 5.5 illustrates the average number of *Interest* packets sent and *Data* packets received, per Consumer.

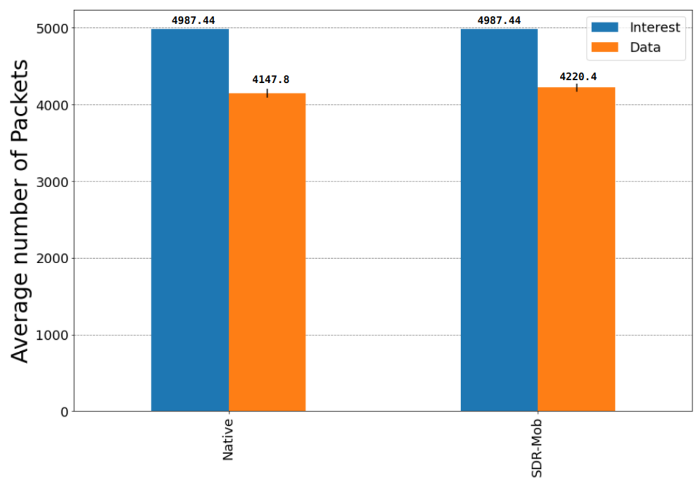


Figure 5.5: Average number of *Interest* packets transmitted and number of *Data* packets received, per mobile Consumer - *Single-Data-Request* functional baseline scenario.

The results show that both native and SDR-Mob deliver approximately in average the same amount of *Data* packets by issuing the same number of *Interest* packets, with the proposed solution with a superior difference of 72.6. To assure the efficiency of the solution, we will rely on the average number of Timeouts regarding the mobile Consumers point of view.

Table 5.2 shows the average number of Consumer Timeouts for both solutions. Through the observed results, it is fair to assume that the SDR-Mob behaves better in a Consumer point of view, reducing 38.86% of the average Timeouts regarding the native approach. This means that throughout the simulation, our solution had more satisfied *Interests* than the native approach.

Table 5.2: Average number of Timeouts per mobile Consumer - *Single-Data-Request* functional baseline scenario.

Solution	Average number of Timeouts
SDR-Mob	318.12 ± 39.93
Native	520.36 ± 48.31

The difference is more visible when evaluating the number of Timeouts rather than the number of received *Data* packets from a Consumer's perspective. The explanation that supports this occurrence is the following: a node can send in a short period of time *Interests* with the same name, having pending those number of requests. That being said, those numerous requests with the same name can be satisfied by the reception of a single content packet with the aforementioned name. Otherwise, if no content with the correspondent name is received, the number of Timeouts are summed by the number of unsatisfied requests (independently of having the same name or not).

Table 5.3 summarizes the performance of all solutions regarding the Consumer satisfaction ratio. It shows the total number of *Interest* and *Data* packets, transmitted and received by all Consumers and Producers. The SDR-Mob achieves a higher Consumer satisfaction ratio, 84.62%, which is 1.45% more than the native approach. This is due to the support of mobility awareness.

Table 5.3: Consumer mobility content retrieval metrics - *Single-Data-Request* functional baseline scenario.

Solution	Packets	Consumers	Producer	Consumer satisfaction ratio
SDR-Mob	*Interest*	124 686	37 932	84.62%
	Data	105 510	37 918	
Native	*Interest*	124 686	42 312	83.17%
	Data	103 696	42 289	

Table 5.4 shows the total and average number of packets transmitted by nodes of the wired infrastructure. It shows the degree of packets travelling in the infrastructure, allowing a comparison between both native and the proposed approaches, and hence determine which one has a bigger network overhead.

From the aforementioned table, we can observe that the SDR-Mob is the one presenting a significant network overhead. If we take into account the transmitted IS as transmitted *Interests*, and the same with DS as *Data*, the total number of transmitted packets from the nodes of the wired infrastructure would be equal to 1 275 207 *Interests* and 1 941 833 *Data* packets for the mobility awareness scenario. On the other hand, the native approach has registered overall a transmission of 358 101 *Interest* packets and 493 937 *Data* packets. This

Table 5.4: Network metrics - *Single-Data-Request* functional baseline scenario.

Solution	Packets	Total number of transmitted packets	Average number of transmitted packets
SDR-Mob	*Interest*	938 586	17 709.16
	Interest Switch	336 621	6 351.34
	Data	1 485 682	28 031.73
	Data Switch	456 151	8 606.60
Native	*Interest*	358 101	6 756.60
	Data	493 937	9 319.56

represents 28.08% and 25.44% of the *Interest* and *Data* packets of the proposed approach, respectively. The significant amount of transmitted packets by the SDR-Mob is not only due to the bifurcations of DS packets into consequent DS packets and *Data* packets, but also because of the interchange communication between RSUs and the MM: (1) the periodic update of neighbor nodes information from each RSU to the MM; (2) and the request of mobility prediction by the RSU to the MM, for each received *Interest* at the respective RSU coming from a mobile Consumer (these dedicated interactions between the RSU and the MM were already explained in Chapter 3). Since the Consumer's *Interest* frequency is equal to 5 packets/s (which means, 5 mobility prediction requests per second only regarding a Consumer), it is not surprising the higher values of *Interest* and *Data* packets interchanged in the SDR-Mob. Further in Section 5.2.2, it is presented and evaluated the *Single-Data-Request* with mobility awareness for higher latency values.

In the following, we study how the number of nodes, *Interest* frequency, wireless range, mobility speed, link delay and Producer delay impacts both native and mobility awareness approaches.

Node's speed range

To evaluate the impact of the node's speed, new scenarios with four different speed ranges were simulated: [5 - 15] units/s, [20 - 30] units/s, [45 - 55] units/s, [95 - 105] units/s. The mobility traces represent a sequential connected straight trajectories, representing the mobility of the node. Each straight trajectory has a speed uniformly distributed in the speed range set.

Figure 5.6 illustrates the average number of Timeouts for both native and SDR-Mob solutions, regarding the different speed ranges. Regarding the native approach, the greater the speed of the node, the greater is the number of Timeouts in a Consumer point of view. This is expected since significant speed allows for more sudden handovers. Hence, the content might not be given soon enough before the Consumer's hand-off. On the other hand, the mobility-aware approach is prepared to handle communication regarding any of the selected speed ranges, deviating the contents appropriately to the nodes' *next RSUs*, since the number of Timeouts does not significantly change with the increase of the nodes' speed.

We can also confirm the consistency of the SDR-Mob regarding the different speed ranges by checking the values obtained in Consumer satisfaction ratio in Table 5.6, whose values are in between 84.46% and 84.73%.

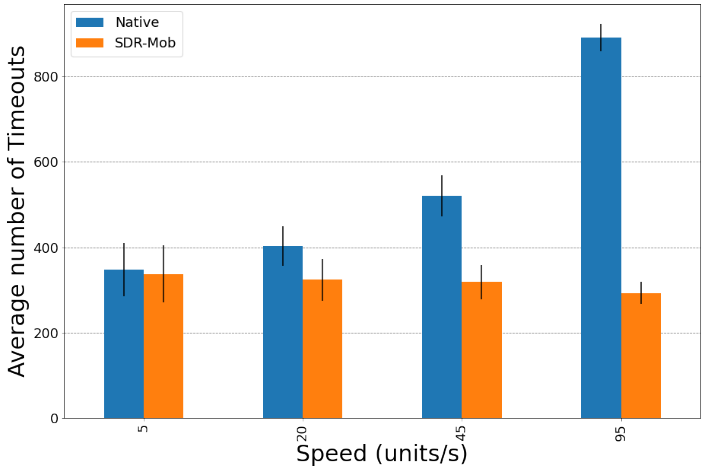


Figure 5.6: Average number of Timeouts regarding the speed of the mobile nodes - *Single-Data-Request* functional scenarios.

Table 5.5: Speed impact in Consumer mobility - *Single-Data-Request* functional scenarios using mobility awareness.

Speed range (units/s)	Packets	Consumers	Producer	Consumer satisfaction ratio
[5 - 15]	*Interest*	124 686	45 476	84.46%
	Data	105 305	45 457	
[20 - 30]	*Interest*	124 686	42 157	84.73%
	Data	105 649	42 130	
[45 - 55]	*Interest*	124 686	37 932	84.62%
	Data	105 510	37 918	
[95 - 105]	*Interest*	124 686	43 657	84.68%
	Data	105 590	43 634	

Number of mobile nodes

To evaluate the impact of the number of nodes, new scenarios were simulated with three different numbers of mobile nodes: 15, 25 and 50. Figure 5.7 illustrates the average number of Timeouts per Consumers in both native SDR-Mob solutions, regarding the different number of mobile nodes.

The results show that the larger the number of mobile nodes, the more notable are the overall Timeouts, for both approaches. This is due to the following: the number of Timeouts can be proportional to the number of deployed *Interest*, since each one represents an *Interest* packet sent from a user. Furthermore, the more users the network has with the same *Interest* rate, the more are the inquiries made. Moreover, the mobile node can have situations of no RSU coverage, making the transmission of *Interests* useless and contributing to the rise of Timeouts throughout the simulation (since it is not satisfied). On top of that, wireless collisions are possible to happen due to the simultaneous transmission of packets in the wireless communication environment. With more packets being transmitted at the same transmission area, the greater are the chances for collisions to happen, and thus originating unsatisfied *Interests*. Despite this, the SDR-Mob solution presents less Timeouts than the native approach.

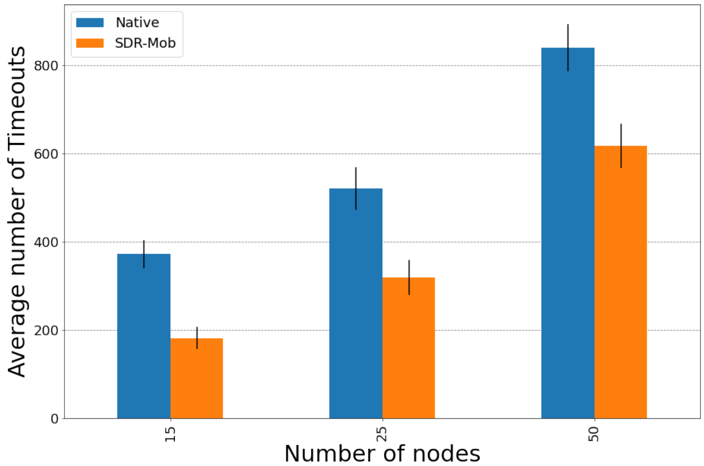


Figure 5.7: Average number of Timeouts regarding the number of mobile nodes - *Single-Data-Request* functional scenarios.

Table 5.6 depicts a degradation of Consumer satisfaction ratio when the number of mobile nodes increases, from a Consumer satisfaction ratio of 87.89% when simulated with 15 nodes, to 77.03% with 50 mobile Consumers. As referred previously in the case of proportional increase of average Timeouts by the number of mobile nodes, this degradation is at least partly due to the collision of packets in the wireless environment.

Table 5.6: Number of mobile nodes impact in Consumer mobility - *Single-Data-Request* functional scenarios using mobility awareness.

Number of mobile nodes	Packets	Consumers	Producer	Consumer satisfaction ratio
15	*Interest*	74 831	24 997	87.89%
	Data	65 771	24 990	
25	*Interest*	124 686	37 932	84.62%
	Data	105 510	37 918	
50	*Interest*	249 215	67 164	77.03%
	Data	191 981	67 123	

Interest frequency

To evaluate the impact of the *Interest* frequency, three different frequencies are considered: 1 packet/second, 5 packets/second and 10 packets/second. Figure 5.8 illustrates the average number of Timeouts for Consumer in both native SDR-Mob approaches, according to their respective *Interest* frequency value.

It is observed an increase in the average Timeouts with the increase of *Interest* frequency; however, with more favourable results in the mobility-aware approach (less average Timeouts than the native approach).

Regarding the results of the SDR-Mob approach, represented in Table 5.7, it is also noticed a degradation of the Consumer satisfaction ratio when the *Interest* frequency increases, from

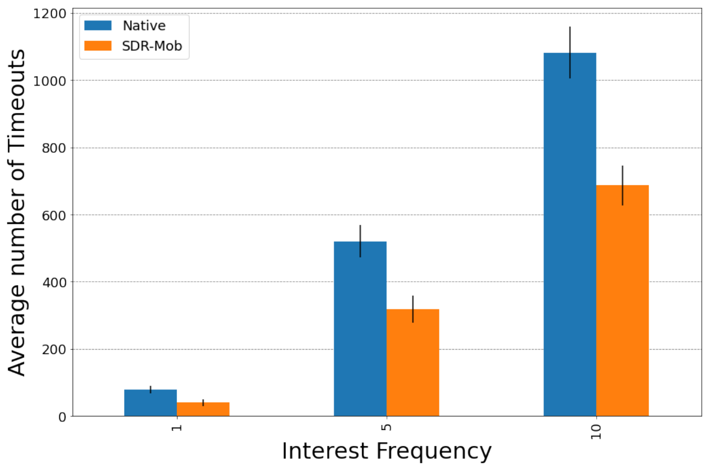


Figure 5.8: Average number of Timeouts regarding the *Interest* frequency - *Single-Data-Request* functional scenarios.

94.31% Consumer satisfaction ratio when the *Interest* frequency is set to 1 packet/s, to 77.24% when the frequency is 10 packet/s. This degradation is at least also partly due to: the more *Interests* are transmitted in the wireless environment, the greater the number of unsatisfied *Interests* due to collision.

Table 5.7: *Interest* frequency impact in Consumer mobility - *Single-Data-Request* functional scenarios using mobility awareness.

***Interest* frequency [packets/s]**	**Packets**	**Consumers**	**Producer**	**Consumer satisfaction ratio**
1	*Interest*	24 945	10 424	94.31%
	Data	23 525	10 419	
5	*Interest*	124 686	37 932	84.62%
	Data	105 510	37 918	
10	*Interest*	249 359	63 631	77.24%
	Data	192 597	63 585	

Producer delay

To evaluate the impact of the content provider delay, four different Producer delays are considered: 100 ms, 500 ms, 1 s and 1.5 s. Figure 5.9 illustrates the average number of Timeouts for the Consumer in both approaches, according to the delay of Producers.

Regarding the native approach, the greater the delay of the Producer is, the greater is the number of Timeouts in a *Consumer's* point of view. This is expected because the RTT is larger than the rest of the time the mobile node is inside the transmission area of the RSU which the *Interest* was requested from. Thus, the content does not arrive soon enough for the Consumer to retrieve it. In the other hand, and only regarding the scenarios with Producer delay equal to 100 ms, 500 ms and 1000 ms, they are prepared to handle communication with such Producer delays using the SDR-Mob, transmitting the contents appropriately to the mobile nodes' *next RSUs*, since the number of Timeouts does not significantly vary with the increase of the Producer delay, along with better results compared to the native approach.

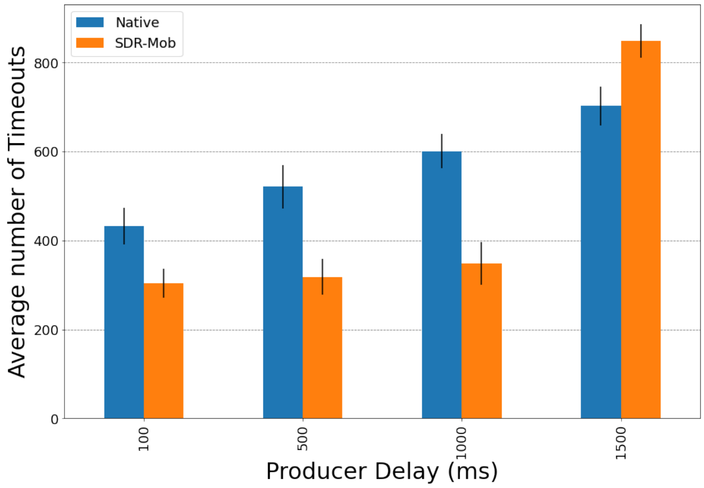


Figure 5.9: Average number of Timeouts regarding the Producer delay - *Single-Data-Request* functional scenarios.

The mentioned consistency reflects the obtained results for Consumer satisfaction ratio represented in Table 5.8. These values oscillate between 84.27% and 85.25%. It is also observed, for the same scenarios, that the Consumer satisfaction ratio is disproportional to the Producer delay, having a Consumer satisfaction ratio of 85.25% for Producer delay equal to 100 ms, and 84.27% when the delay is equal to 1 second.

Table 5.8: Producer delay impact in Consumer mobility - *Single-Data-Request* functional scenarios using mobility awareness.

Producer delay [ms]	Packets	Consumers	Producer	Consumer satisfaction ratio
100	*Interest*	124 686	36 389	85.25%
	Data	106 292	36 384	
500	*Interest*	124 686	37 932	84.62%
	Data	105 510	37 932	
1000	*Interest*	124 686	38 954	84.27%
	Data	105 067	38 917	
1500	*Interest*	124 686	39 538	83.78%
	Data	104 467	39 469	

Although our solution presents better results than the native approach for scenarios using a Producer delay of 100 ms up to 1 s, when considering 1.5 s, the SDR-Mob obtains a larger number of Consumer Timeouts, as can be observed in Figure 5.9. The explanation for this inappropriate behaviour is the following: in the scenario using mobility awareness, a content request from the mobile Consumer involves two sub-sequential communications, which are the mobility prediction request (communication between RSU and MM) and then the actual content retrieval, which could imply content bifurcation if specified, in order to reach the Consumer at its new position. That being said, there are two RTTs involved for an *Interest* to be satisfied. Accumulating both of these times, mainly influenced by the imposed link delay and Producer delay, the time it takes the actual content to finally arrive to the mobile

Consumer surpasses the *Interest* lifetime and configured Timeout of the request, which is 3 seconds, as referred in Table 5.1. This problem could be tackled by modifying the sessions responsible to communicate the mobility predictions, further detailed in Section 5.2.2.

Link delay

To evaluate the impact of the content provider delay, it is considered four different link delays: 10 ms, 50 ms, 120 ms and 160 ms. Figure 5.10 illustrates the average number of Timeouts per Consumer in both approaches, for different link delays of the wired infrastructure.

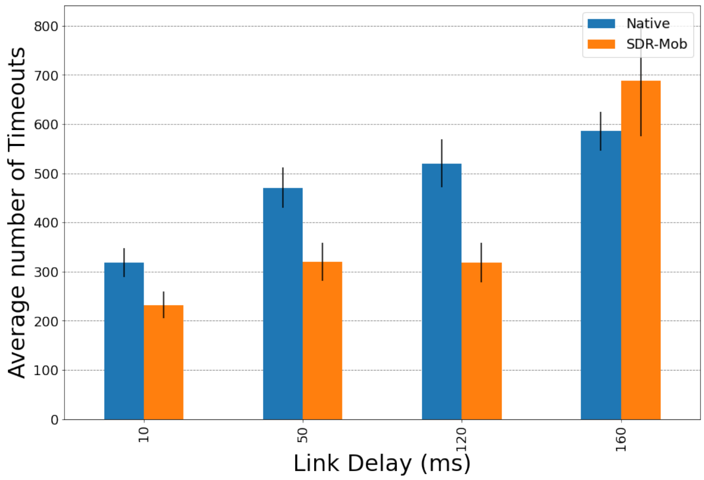


Figure 5.10: Average number of Timeouts regarding the links delay - *Single-Data-Request* functional scenarios.

The behaviour observed in Figure 5.10 is the same as the one expressed in the case of the evaluated scenarios for different Producer delays: Timeouts increase with the link delay for the native solution, due to the potential occurrence of hand-off before the content arrives to the RSU, since the slow interchange of *Interest* and consequent *Data* packet, whose delay is the result of the several link delays which the packets traverse. Moreover, for the most parts (link delay equal to 10 ms, 50 ms and 120 ms), the SDR-Mob manages to have a constant average number of Timeouts, smaller than the native approach. This is because the topology is ready to deliver the content to the RSU reflecting the position of the mobile content requester.

Just as in the scenario with Producer delay equal to 1.5 s, the SDR-Mob presents worst results when the link delay (in this case, a link delay of 160 ms) is large enough for the content request timeout to be surpassed before it actually arrives to the Consumer (which is not acknowledged and processed by the Consumer application, since the PIT entry of the node regarding the request is also dropped due to timeout). In Section 5.2.2, this issue will be tackled with a different *Single-Data-Request* approach with mobility awareness as well, decreasing the number of Timeouts when the link delays are high enough to cause packet loss.

Finally, considering the SDR-Mob, in Table 5.9 it is noticeable a degradation of the Consumer satisfaction ratio with the link delay, from 92.72 % with 10 ms of link delay, to 82.65 % when the link delay is equal to 160 ms.

Table 5.9: Link delay impact in Consumer mobility - *Single-Data-Request* functional scenarios using mobility awareness.

Delay link [ms]	Packets	Consumers	Producer	Consumer satisfaction ratio
10	*Interest*	124 686	42 860	92.72%
	Data	115 610	42 842	
50	*Interest*	124 686	40 138	87.88%
	Data	109 569	40 110	
120	*Interest*	124 686	37 932	84.62%
	Data	105 510	37 918	
160	*Interest*	124 686	36 663	82.65%
	Data	103 059	36 646	

Wireless range

To evaluate the impact of the wireless range of the RSUs, three different radius regarding the circular transmission coverage are considered: 100 units, 120 units and 150 units. Figure 5.11 illustrates the average number of Timeouts per Consumer in both native and SDR-Mob approaches, for different wireless ranges.

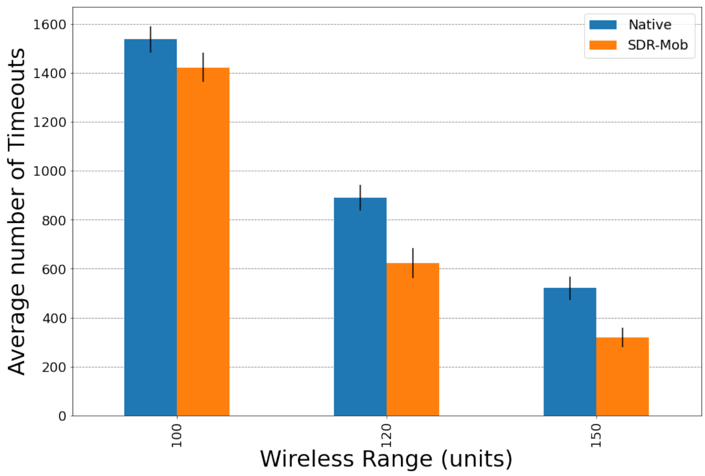


Figure 5.11: Average number of Timeouts regarding the wireless range of the RSUs - *Single-Data-Request* functional scenarios.

Figure 5.11 shows, for both approaches, that the obtained average number of Timeouts are inversely proportional to the wireless range of the RSUs. The smaller the coverage area is, the more prone are the nodes to hand-off. Also, the smaller the radius, the smaller is the intersection of transmission areas between adjacent RSUs. In these areas, the mobile node can request the content from all the RSUs it has coverage, being delivered to the Consumer from both RSUs or one of the RSU it transitioned to. For these situations, it would not be required the use of our approach to handle handover, since the Consumer made the request from the current RSU and the next one which is about to migrate. Hence, with small areas, these areas would become smaller or even non-existent, and therefore, the node would not have great chances to received the content to the RSU it is going to transition to without the employment of our approach.

Furthermore, small RSU wireless ranges lead to an increase of areas without communication coverage. This happens in the scenarios with wireless ranges equal to 120 and 100 units. Moreover, the SDR-Mob approach presents significantly smaller number of Timeouts, compared to the native approach, regarding the chosen wireless ranges, since it handles Consumer mobility. The values from Figure 5.11 reflect the Consumer satisfaction ratio results shown in Table 5.10: large wireless transmission area results in higher Consumer satisfaction ratios than smaller ones. The Consumer satisfaction ratio increases from 58.86% with wireless radius range of 100 units, to 84.62 % when the later is configured with 150 units.

Table 5.10: Wireless range impact in Consumer mobility - *Single-Data-Request* functional scenarios using mobility awareness.

Wireless range [units]	**Packets**	**Consumers**	**Producer**	**Consumer satisfaction ratio**
100	*Interest*	124 686	20 087	58.86%
	Data	73 395	20 073	
120	*Interest*	124 686	26 895	77.39%
	Data	96 495	26 880	
150	*Interest*	124 686	37 932	84.62%
	Data	105 510	37 918	

Scenarios with real vehicular mobility traces

This section analyses the proposed SDR-Mob approach behaviour in Consumer mobility scenarios with real mobility traces, by comparing it with the NDN native strategy. From the real data, we selected 25 mobile nodes to act as Consumers, and three static Producers. The MM responsible for forecasting the node's *next RSUs* is manufactured in the pipeline mentioned in Section 5.1.4, having historical data about connectivity of each mobile Consumer with the infrastructure's RSUs, throughout time. The configured time window to check the Consumers' *next RSUs* in the next instants is equal to 3 seconds. The remaining configurations are illustrated in Table 5.11. The values in this section are not cumulative, but per Consumer, and therefore, they represent the mean and 95% confidence intervals.

Table 5.11: Baseline simulation configurations - *Single-Data-Request* scenario using real vehicular mobility traces.

Average velocity	50 $\pm 10 units/s$
Number of mobile Consumers	25 nodes
Delay link	120 ms
Number of existing contents per prefix	70
Interest frequency	0.5 packets/s
Wireless range	1000 units
Producer delay	1500 ms
Content Store size	30

To evaluate the base scenario, for both native and SDR-Mob approaches, with real mobility traces, it is presented the following: Figure 5.12, illustrating the average number of *Interest* and

Data packets transmitted and received by the mobile Consumers, respectively, and Table 5.12, showing the average and total number of content Timeouts registered at the Consumer.

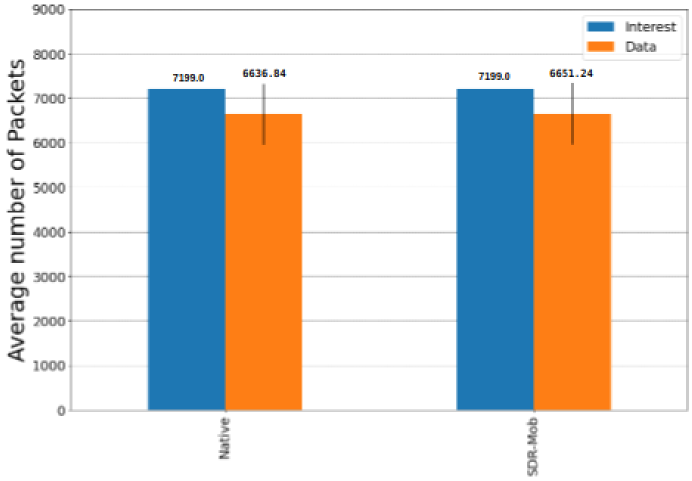


Figure 5.12: Average number of *Interest* packets transmitted and number of *Data* packets received, per mobile Consumer - *Single-Data-Request* baseline scenario using real vehicular mobility traces.

Table 5.12: Timeout results per mobile Consumer - *Single-Data-Request* scenarios using real vehicular mobility traces.

Solution	**Total number of Timeouts**	**Average number of Timeouts**
SDR-Mob	11 790	471.60 ± 647.43
Native	12 253	490.12 ± 646.03

From the results shown in Figure 5.12 and Table 5.12, it is easily observed a resemblance between both approaches: the average and total number of Timeouts are similar between the two approaches, having a difference of 18.5 average Timeouts and a total number of 463 Timeouts from the mobile Consumers' perspective, showing better results when using the mobility awareness approach. Furthermore, the total number of *Data* packets received by the Consumers in both approaches is very close (the native approach registers 6 636.84 average number of packets, and the SDR-Mob registers 6 651.24). This closeness is also noticeable in the results regarding Consumer satisfaction ratio, presented in Table 5.13. The native approach has a Consumer satisfaction ratio of 92.19%, and the one using the proposed scheme has 93.39%, having between them a difference of 1.20%. However, notice the large variation on the average number of Timeouts, which requires caution when providing conclusions.

Despite the SDR-Mob scheme being able to achieve slightly better results for this particular scenario setup, it might not be appropriate to use it when the introduction of overhead in the infrastructure is crucial. In Table 5.14, it is easily observed that the average number of *Interests* transmitted by the elements of the wired infrastructure in the SDR-Mob approach almost doubles the number of regular *Interests* of the native approach. Moreover, the average number of *Data* packets increases from 10 577.03 to 17 735.34, from the native approach to the

Table 5.13: Consumer mobility content retrieval metrics - *Single-Data-Request* baseline scenario using real vehicular mobility traces.

Solution	Packets	Consumers	Producer	Consumer satisfaction ratio
SDR-Mob	*Interest*	179 975	64 657	93.39%
	Data	166 281	64 651	
Native	*Interest*	179 975	56 642	92.19%
	Data	165 921	56 636	

one proposed. This increase in the average packet transmission has already been explained in Section 5.2.1, more specifically when evaluating the overhead of the base functional scenarios using both schemes. On top of that, the proposed solution causes additional overhead coming from the introduced support packets (IS and DS packets): each node from the wired topology sends an average of 1 419.62 IS and 1 418.63 DS packets.

Table 5.14: Network metrics - *Single-Data-Request* baseline scenario using real vehicular mobility traces.

Solution	Packets	Total number of received packets	Average number of received packets
SDR-Mob	*Interest*	1 112 187	15 235.44
	Interest Switch	103 632	1 419.62
	Data	1 294 680	17 735.34
	Data Switch	103 560	1 418.63
Native	*Interest*	504 392	6 909.48
	Data	772 123	10 577.03

Because of the proximity of the results obtained regarding Timeouts and Consumer satisfaction ratio between both approaches, it is not discarded the hypothesis of collision of packets in an wireless environment to influence the results regarding the metrics aforementioned. That being said, we proceed on increasing the Producer delay to different values in order to understand if the proposed approach is suitable for the scenarios using the acquired real vehicular mobility traces. This way, we force the situations where Consumers would handover before obtaining the requested content. The additional selected Producer delay values were 5 s, 10 s, and 120 s. For these scenarios, we also changed the Timeout timer, taking into account the Producer delay, to avoid drop of packets due to overpass the *Interest* lifetimes of the requests, as it happened in the case of the functional scenario with Producer delay of 1500 ms, which timeout was settled to 3 seconds, from Section 5.2.1.

Figure 5.14 shows the average number of received *Data* packets, according to the several Producer delays.

In this figure, focusing on the scenarios using 1.5 seconds, 5 seconds and 10 seconds, it is shown almost no significant difference between the number of *Data* packets received by the Consumer using a NDN native approach and the proposed scheme. On top of that, in the case where the link delay is equal to 10 seconds, the proposed approach presents an average of received *Data* packets equal to 6 504.56, which is 21.07 less than the average of received

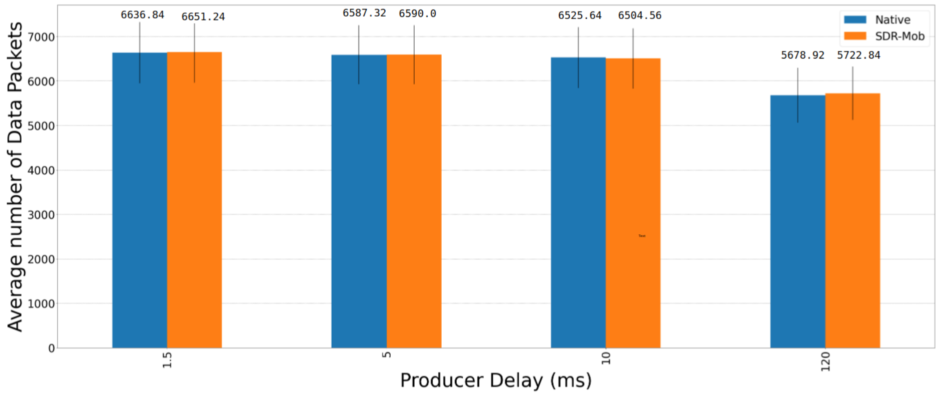


Figure 5.13: Average number of Timeouts regarding the Producer delay - *Single-Data-Request* scenarios using real vehicular mobility traces.

packets by the native approach. This re-enforces the aforementioned hypothesis that the scenarios using the proposed scheme behave slightly better than using the native approach because of causality. It is observed in Table 5.15, that despite cases with Producer Delay 1.5 seconds and 5 seconds has slightly less number of Timeouts using the SDR-Mob scheme, in the case where it is employed link delay equal to 5 seconds, the scenario using NDN native approach behaves better in a Consumer perspective, achieving less 206 Timeouts.

Finally, it was simulated a topology with a link delay of 120 seconds. For this scenario, a Consumer would have, in average, almost 50 more *Data* packets received with the proposed scheme, as denoted in Figure 5.14. Furthermore, we can observe from Table 5.15 that the solution has less 1 461 Timeouts than the one obtained by the native approach. Despite seeming appropriate to use the SDR-Mob scheme in high latency scenarios, along with the support of appropriate *Interest* lifetime (otherwise PIT entries would be dropped), this might not be practical and feasible: nowadays it is unacceptable for services and delivery of content to have latencies of such degree, specially with the advances of communication technologies, such as optical fiber and servers with a high level of performance.

Table 5.15: Total number of Timeouts per Producer delay. Impact in Consumer mobility - *Single-Data-Request* solution using real vehicular mobility traces.

Producer delay [s]	Native	SDR-Mob
1.5	12 253	11 790
5	11 877	11 319
10	11 296	11 501
120	7 633	6 172

Unlike the functional scenarios from Section 5.2.1, the scenarios using real vehicular mobility traces are not getting significant benefits from the proposed schemes to support Consumer mobility in situations of handovers. The velocity of the nodes throughout simulation are enough for them to retrieve the content from the same RSUs where the requests were sent to the ISP. If the transition from one RSU transmission area to another was as fast as the one presented in functional scenarios, the proposed Consumer mobility scheme would have presented a better quality of service than the native approach.

5.2.2 *Single-Data-Request* with notification

In Section 5.2.1 it has been noticed that the scheme proposed for *Single-Data-Request* damages the expected content reception: the time it takes for the content to reach the Consumer surpasses the timeout timer of the content request. This time depends on the RTT related to the retrieval of mobility prediction and the RTT related to the content acquaintance for the Consumer to its location. By having proactive retrieval of predictions from the MM, and having the RSUs to persist information of the potential *next RSUs* of their neighbors, the content retrieval time would only be dependent on the RTT related to the content retrieval. This is, an IS is sent to the ISP if the RSU has already been informed about the potential migration of the respective neighbor Consumer. Otherwise, it sends a regular *Interest* to retrieve the desired content.

Moreover, it was also observed, from Table 5.4, that there is a significant network overhead cost to support the mobility-aware approach, in comparison to the NDN native approach. This was mostly due to the communication established between RSUs and MM, for the communication of mobility predictions of mobile devices, established every time an RSU received an *Interest* packet coming from a mobile Consumer.

This section evaluates a solution where the mobility prediction is obtained through the use of notifications, denoted as SDR-Mob-N. It is also presented the results obtained with the NDN native solution and the original proposed scheme with mobility awareness (SDR-Mob), for comparison purposes. Finally, it is also evaluated the overall overhead caused at the entities from the wired infrastructure and their efficiency in a mobile Consumer perspective.

The base scenario configurations are the same as the one stated in Section 5.2.1, and Table 5.1. Regarding the MM, which is obtained from the pipeline explained in Section 5.1.4, it is implemented a scheduler in which for each 30 seconds, it is determined for the mobile Consumers with coverage from at least one RSU (this information is determined by the neighborhood information given from the RSUs to the MM, using pub-sub), which are the mobile node's *next RSUs* for the next 30 seconds (windows equal to 30 seconds). Notice that the values in this section are not cumulative, but per Consumer, and therefore they represent the mean and 95% confidence intervals.

Figure 5.14 illustrates the average number of *Interest* packets sent and *Data* packets received, per Consumer. The results show that the three solutions deliver approximately the same amount of *Data* to the Consumers, issuing the same number of *Interest* packets. These values are also reflected in the Consumer satisfaction rate results shown in Table 5.16. To analyse the efficiency in the solutions supporting Consumer mobility, we additionally analyze, in Table 5.17, the average number of Timeouts issued by the mobile Consumers.

In Table 5.17 it is observed that both mobility awareness schemes help to reduce the average number of Timeouts registered at the Consumers. However, the SDR-Mob-N does not reduce the Timeouts as the SDR-Mob, having a difference of almost 100 Timeouts less in average. The regular mobility-aware approach determines the *next RSUs* per Consumer

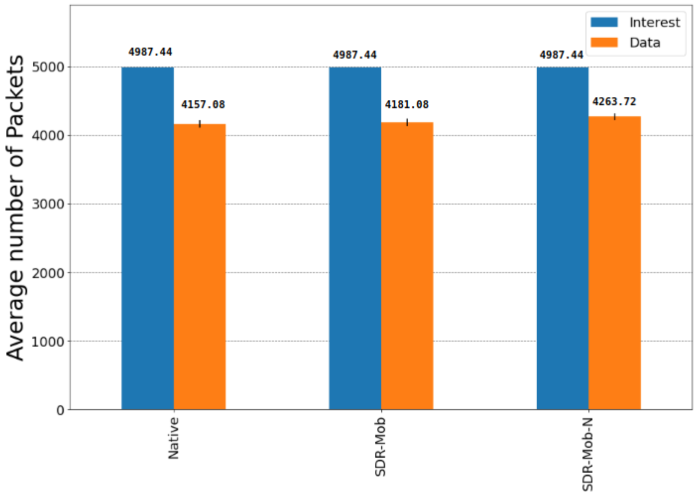


Figure 5.14: Average number of *Interest* packets transmitted and number of *Data* packets received, per mobile Consumer - *Single-Data-Request* functional baseline scenario with native, SDR-Mob and SDR-Mob-N solutions.

Table 5.16: Consumer mobility content retrieval metrics - *Single-Data-Request* functional baseline scenario with native, SDR-Mob and SDR-Mob-N solutions.

Solution	Packets	Consumers	Producer	Consumer satisfaction ratio
SDR-Mob-N	*Interest*	124 686	42 877	83.83%
	Data	104 527	42 848	
SDR-Mob	*Interest*	124 686	42 358	85.49%
	Data	106 593	42 342	
Native	*Interest*	124 686	42 468	83.35%
	Data	103 927	42 451	

Interest transmitted, therefore a more strict prevision and settlement of new data transmission path is done to reach the Consumers new position, when a potential handover is happening. Regarding the proposed notification scheme, the settlement is only done when the RSU receives a notification about potential migrations. On top of that, each RSU must first inform the MM, through the periodic neighbor updates about its nearby mobile Consumers (which is done each 10 seconds). Once the MM knows about the actual mobile Consumer neighbors of the infrastructure's RSUs, it can then calculate and inform about potential migrations of these Consumers to their respective RSUs that are hosting them. Otherwise, without this contextualization from the MM, no prediction, no notification and therefore no handling of content transmission to the next RSUs reflecting the mobile node's next position can be established.

Despite not being as efficient as SDR-Mob, the SDR-Mob-N alleviates significantly the overhead at the elements from the wired infrastructure as observed in Table 5.18, showing the total and average number of transmitted packets by the elements of the wired topology (from an average of 18 520 transmitted *Interests* and 26 127 transmitted *Data* packets with the SDR-Mob to 11 591 transmitted *Interests* and 14 144 transmitted *Data* packets, using

Table 5.17: Average number of Timeouts per mobile Consumer - *Single-Data-Request* functional baseline scenario with native, SDR-Mob and SDR-Mob-N solutions.

Solution	**Average number of Timeouts**
SDR-Mob-N	426.68 ± 40.78
SDR-Mob	337.80 ± 44.87
Native	511.76 ± 43.82

the SDR-Mob-N). It is also observed a significant reduction of the IS and DS, 80.28% of transmitted IS and 60.51% of transmitted DS, from the regular mobility awareness solution to the similar solution with notifications. The employed number of IS and DS, although it helps to diminish the number of Timeouts with the actual transmission of content to the *next RSUs*, they are not enough to obtain the same number of Timeouts as the ones obtained when mobility prediction is requested per Consumer *Interest* at the RSUs.

Table 5.18: Network metrics - *Single-Data-Request* functional baseline scenario with native, SDR-Mob and SDR-Mob-N solutions.

Solution	**Packets**	**Total number of received packets**	**Average number of Received packets**
SDR-Mob-N	*Interest*	614 337	11 591.00
	Interest Switch	58 919	1 111.68
	Data	749 667	14 144.66
	Data Switch	144 744	2 731.02
SDR-Mob	*Interest*	981 052	18 510.42
	Interest Switch	298 749	5 636.77
	Data	1 384 748	26 127.32
	Data Switch	366 545	6 915.94
Native	*Interest*	362 993	6 848.92
	Data	499 070	9 416.42

Finally, Figures 5.15 and 5.16 show that, when using 1500 ms of Producer delay and 160 ms of link delay, respectively, the SDR-Mob approach substantially surpasses the average number of Timeouts of the ones obtained by the NDN native approach. However, the SDR-Mob-N provides Consumer mobility notification and succeeds to diminish the average number of Timeouts from the native approach.

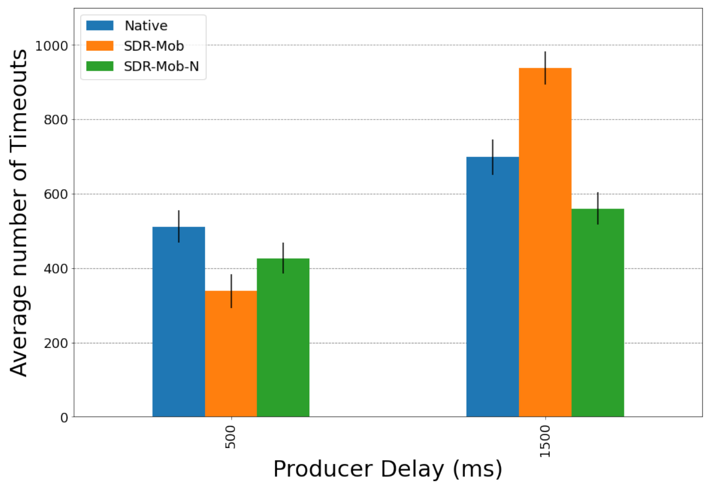


Figure 5.15: Average number of Timeouts regarding the Producer delay - *Single-Data-Request* functional baseline scenario with native, SDR-Mob and SDR-Mob-N solutions.

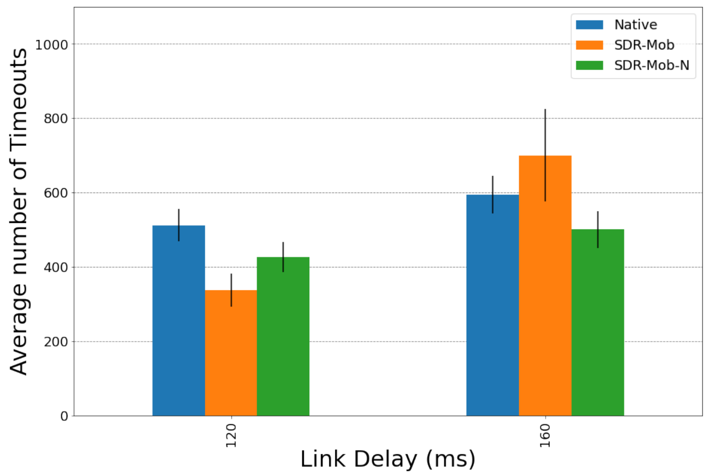


Figure 5.16: Average number of Timeouts regarding the link delay - *Single-Data-Request* functional baseline scenario with native, SDR-Mob and SDR-Mob-N solutions.

5.2.3 Pub-sub communication model

Functional scenarios

This section analyses the pub-sub solution with mobility awareness, hereafter simply denoted as PubSub-Mob, in a Consumer mobility scenario, by comparing it with a simple pub-sub solution without mobility awareness, simply denoted by PubSub. After a general analysis between the two solutions, we study how a different number of nodes, speed range and wireless communication range impact the aforementioned solutions. For these scenarios it is employed three Producers. The MM responsible for forecasting the node's mobility is obtained by the pipeline from Section 5.1.4. The remaining configurations are illustrated in Table 5.19, and the topology is similar to the one in the *Single-Data-Request*.

Figure 5.17 illustrates the average number of *Interest* packets sent and *Data* packets received. When comparing both pub-sub solutions, the PubSub-Mob is the one capable of

Table 5.19: Baseline simulation configurations - pub-sub functional scenario.

	PubSub	PubSub-Mob
Simulation duration	1000s	
Number of Publishers	3	
Number of Subscribers	25	
***Interest* lifetime**	30s	
***Data* transmission rate**	2s	

delivering more *Data* packets due to its mobility awareness feature. Moreover, the number of *Interest* subscription packets is significantly lower, causing low network overhead. These results are also reflected in Table 5.20, showing the average number of Timeouts which are substantially lower when using PubSub-Mob solution.

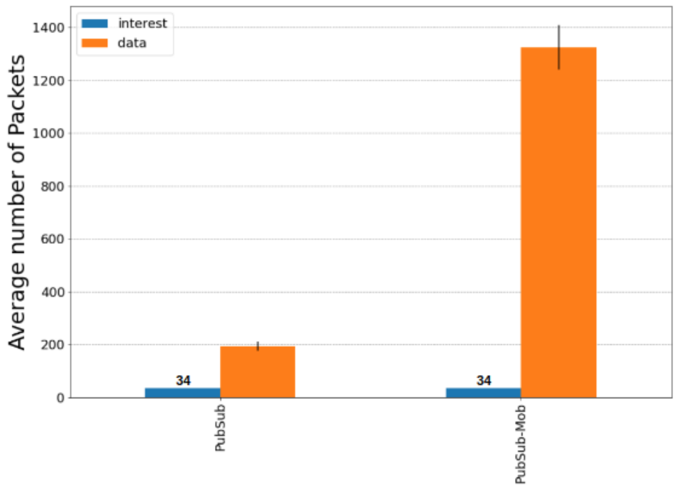


Figure 5.17: Average number of *Interest* packets transmitted and number of *Data* packets received, per mobile Consumer - pub-sub functional baseline scenario.

Table 5.20: Average number of Timeouts per mobile Consumer - pub-sub functional baseline scenario.

Solution	**Average number of Timeouts**
PubSub-Mob	18.68 ± 6.28
PubSub	323.52 ± 17.84

Table 5.21 summarizes the performance of both solutions regarding the Consumer satisfaction ratio. It shows the total number of *Interest* and *Data* packets, transmitted and received by all Consumers and the Producer. Regarding the pub-sub solutions, they present a Consumer satisfaction ratio of 38.6 % and 264.8 %, with and without mobility awareness, respectively. The enormous Consumer satisfaction ratio obtained in PubSub-Mob is justified by the delivery of the same *Data* packet by different RSUs in the coverage of the same mobile Consumer.

Table 5.22 shows the consequent overhead registered by the entities of the wired infrastructure, for both pub-sub solutions. Apart from the introduction of the support packets

Table 5.21: Consumer mobility content retrieval metrics - pub-sub functional baseline scenario.

Solution	Packets	Consumers	Producer	Consumer satisfaction ratio	*Data* received per *Interest* transmitted
PubSub-Mob	*Interest*	850	1 000	264.8%	38.94
	Data	33 098	2 284		
PubSub	*Interest*	850	411	38.6%	5.68
	Data	4 825	1 497		

of SIS and SDS to transmit the content and establish a new persistent pub-sub path from Publisher towards the Consumer's neighbor RSUs reflecting its potential position, the results show that the total and average number of *Interest* and *Data* transmitted from the elements of the wired infrastructure are high when using PubSub-Mob approach: the number of *Interest* increases almost 30% and the number of *Data* packets increases 6 times, from PubSub to PubSub-Mob. The significant amount of transmitted packets by this solution is not only due to the bifurcation of SDS packets into consequent SDS packets and *Data* packets, and the subsequent *Data* packets transmitted through the new paths (until the PIT lifetime expires), but also because of the interchange communication between RSUs and MM: the periodic update of neighbor nodes information from each RSU to the MM, and the notifications about potential migrations from the MM to the RSUs hosting the mobile Consumer.

Table 5.22: Network metrics - pub-sub functional baseline scenario.

Solution	Packets	Total number of received packets	Average number of received packets
PubSub-Mob	*Interest*	8 792	165.88
	Interest Switch	3 635	68.58
	Data	332 010	6 264.33
	Data Switch	16 245	306.50
PubSub	*Interest*	6 391	120.58
	Data	58 320	1 100.37

Number of mobile nodes

To evaluate the impact of the number of mobile nodes, it was simulated scenarios with three different network densities: 15, 25 and 50 mobile nodes.

Figure 5.7 illustrates the average number of Timeouts, for PubSub and PubSub-Mob, for different number of mobile nodes.

Regarding the simple pub-sub solution, the more mobile nodes the infrastructure has, the less average number of Timeouts are registered, passing from an average of 350 Timeouts when there is 25 nodes to 272 when the number of node doubles. This happens because there are mobile nodes consuming the same content. That being said, the more nodes are subscribed to the same topic, the higher are the chances for a node to migrate to another transmission area and have the subscription path already settled by another common passing mobile Consumer. Furthermore, the PubSub-Mob presents a reduced and similar average number of Timeouts between all scenarios, meaning that the solution is capable to afford the content to its Subscribers independently of the adherence of Consumers in the network.

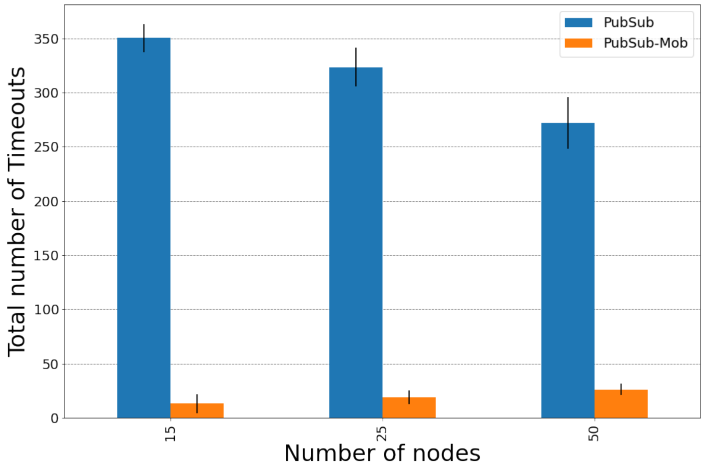


Figure 5.18: Average number of Timeouts regarding the number of mobile nodes - pub-sub functional scenarios.

Table 5.23 sums up the impact of the number of nodes in the PubSub-Mob solution. The first observation goes to the Consumer satisfaction ratio as the number of nodes increases: it increases and reaches 286.6 % for a simulation with 50 nodes, surpassing substantially the required number of expected content throughout the simulation (since it is bigger than 100%). Such behaviour is justified because of the reception of replicated content when it is transmitted by more than one RSU, when in multiple coverage. This is furthermore justified by the reuse of the reverse paths created by the SIS packets used in our solution.

Table 5.23: Number of mobile nodes impact in Consumer mobility - pub-sub functional scenarios using mobility awareness.

Number of mobile nodes	Packets	Consumers	Producer	Consumer satisfaction ratio	*Data* received per *Interest* transmitted
15	*Interest*	510	646	213.1%	31.40
	Data	16 013	1 976		
25	*Interest*	850	975	255.9%	37.64
	Data	31 990	2 257		
50	*Interest*	1 700	1 530	286.6%	42.15
	Data	71 655	2 602		

Speed range

To evaluate the impact of the node's speed, it was simulated four different speed ranges: [5 - 15] units/s, [20 - 30] units/s, [45 - 55] units/s, [95 - 105] units/s.

Figure 5.19 illustrates the average number of Timeouts for scenarios using both PubSub and PusbSub-Mob for different speed ranges.

Regarding the simple pub-sub solution, the faster the nodes are, the higher becomes the average number of Timeouts. This happens because the faster the node goes, the less is the utilization of established content subscription, due to faster handoffs without the support for Consumer mobility. On the other hand, scenarios using PubSub-Mob are capable to maintain

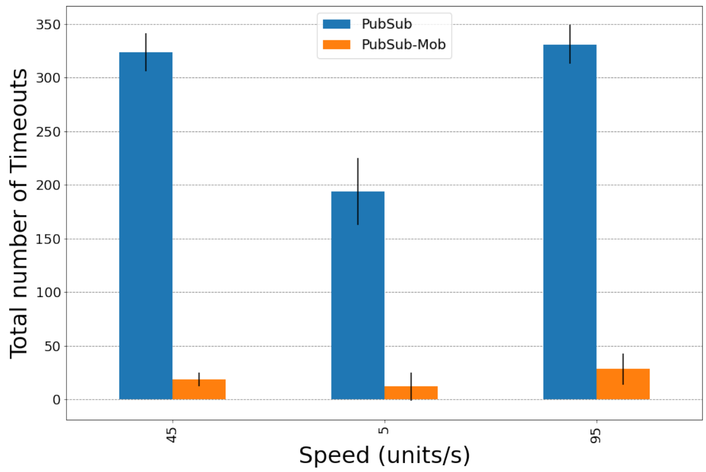


Figure 5.19: Average number of Timeouts regarding the speed of mobile nodes - pub-sub functional scenarios.

the same quality of service independently of the speed of the Consumer, due to the support of the proposed mobility awareness for pub-sub communications.

Table 5.24 sums up the impact of the speed of nodes in the PubSub-Mob solution. Both Consumer satisfaction ratio and *Data* Received per *Interest* transmitted decreases with the increase of Consumer speed. For the slowest scenario, the obtained Consumer satisfaction ratio and *Data* received per *Interest* transmitted is equal to 277% and 40%, respectively. This points out that the slower the node is, the better is the use of the established topic subscriptions, independently of the case if the settlement was established by their own, or by another node who has transitioned to the same coverage area. One should not forget that the Consumer is also prompt of receiving replicated content coming from multiple RSUs in coverage.

Table 5.24: Nodes' speed ranges impact in Consumer mobility - pub-sub functional scenarios using mobility awareness.

Speed range [units/s]	Packets	Consumers	Producer	Consumer satisfaction ratio	*Data* received per *Interest* transmitted
[45 - 55]	*Interest*	850	975	255.9%	37.64
	Data	31 990	2 257		
[5 - 10]	*Interest*	850	1 114	277.9%	40.88
	Data	34 746	2 466		
[95 - 105]	*Interest*	850	864	237.1%	34.88
	Data	29 646	2 071		

Wireless range

To evaluate the impact of the wireless range of the RSUs, it was simulated scenarios using three different RSUs radio coverages: 100 units, 120 units and 150 units. Figure 5.20 illustrates the average number of Timeouts for Consumer in both PubSub and PubSub-Mob solution, for several wireless communications ranges.

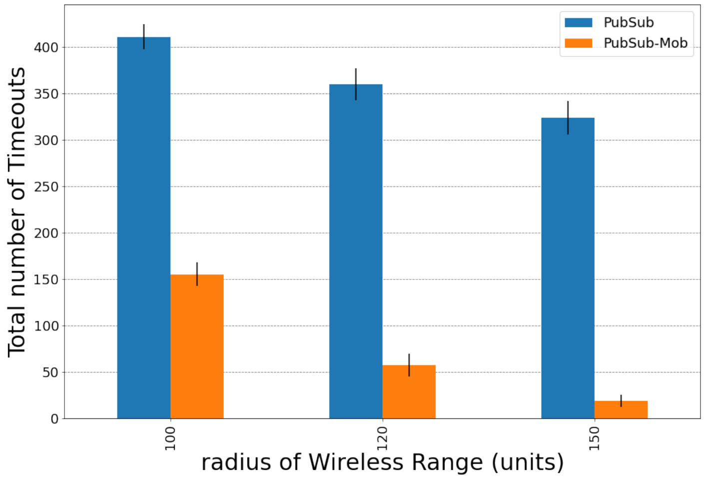


Figure 5.20: Average number of Timeouts regarding the RSUs wireless range - pub-sub functional scenarios.

Figure 5.20 points out, for scenarios using native and proposed mobility awareness approach, that the obtained average number of Timeouts are inversely proportional to the wireless range of the RSUs. The lower the coverage area is, the more prone are the nodes to hand-off. Furthermore, small RSU wireless ranges lead to an increase of areas without communication coverage. Moreover, the proposed solution presents significantly smaller number of Timeouts, compared to the scenarios using PubSub, regarding the chosen wireless range, since it handles Consumer mobility. The obtained values in Figure 5.20 reflect the Consumer satisfaction ratio results shown in Table 5.25: big wireless transmission area causes better Consumer satisfaction ratio results. The Consumer satisfaction ratio increases from 122% with wireless radius range of 100 units to 255% when configured with 150 units.

Table 5.25: Wireless ranges impact in Consumer mobility - pub-sub functional scenarios using mobility awareness.

Wireless range [units]	Packets	Consumers	Producer	Consumer satisfaction ratio	*Data* received per *Interest* transmitted
100	*Interest*	850	633	122.1%	17.96
	Data	15 262	1 926		
120	*Interest*	850	808	195.9%	28.82
	Data	24 493	2 097		
150	*Interest*	850	975	255.9%	37.64
	Data	31 990	2 257		

Link delay

To evaluate the impact of the link delay, four different scenarios were simulated: 10 ms, 50 ms, 120 ms and 160 ms. Figure 5.21 illustrates the average number of Timeouts for both PubSub and PubSub-Mob, with different link delays of the wired infrastructure.

Figure 5.21 shows that link delays do not significantly influence the average number of Timeouts per Consumer, no matter the solution. Moreover, it is observed that the PubSub-Mob clearly outperforms the PubSub with a reduced number of Timeouts.

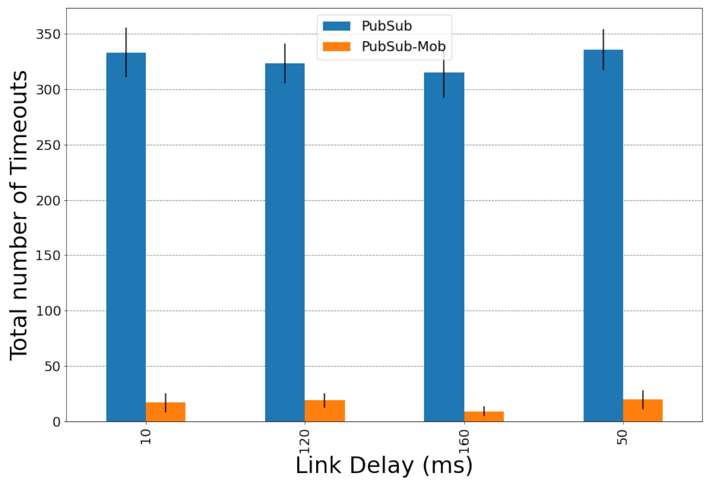


Figure 5.21: Average number of Timeouts regarding the link delay - pub-sub functional scenarios.

Scenarios with real vehicular mobility traces

This section analyzes the proposed PubSub-Mob solution in a Consumer mobility scenario with real vehicular mobility traces. It was randomly selected 25 mobile nodes to act as Consumers, and one static Producer. For this time, we are not resorting to a built MM coming from the pipeline described in Section 5.1.4. Instead, the mobility predictor estimates the vehicle's next position using a Markovian Model, with each RSU reporting its neighboring information every 30 seconds [62]. The accuracy associated with the prediction is around 95%. The remaining configurations are illustrated in Table 5.26. Notice that the values in this section are not cumulative, but per Consumer, and therefore, they represent the mean and 95% confidence intervals.

First, we proceed with a general analysis between three solutions using scenarios with the previous indicated baseline configurations. The solutions are: the NDN native approach, which represents the request of a single *Data* packet through an *Interest*, without Consumer mobility awareness; PubSub, which uses persistent PIT entries through an *Interest* packet transmission for multiple content throughout time, without using Consumer mobility awareness; and finally, our developed PubSub-Mob. After that, we study how different *Data* transmission rates and *Interest* lifetimes impact the pub-sub solutions.

Table 5.26: Baseline simulation configurations - pub-sub scenario with real vehicular mobility traces.

	Native	**PubSub**	**PubSub-Mob**
Simulation duration	14400s		
Number of Publishers	1		
Number of Subscribers	25		
***Interest* lifetime**	2s	600s	
***Data* transmission rate**	N/A	2s	
Expiry event	N/A	N/A	30

Figure 5.22 illustrates the average number of *Interest* packets sent and *Data* packets received, per Consumer. The results show that the native solution is able to deliver a higher

number of *Data* packets when compared to the pub-sub solutions. However, this behavior is achieved at the cost of a significant network overhead, since each *Data* packet demands for an associated *Interest* packet. As for the pub-sub solutions, the number of *Data* packets received by the mobile Consumers is slightly lower, but the number of *Interest* subscription packets is significantly lower, reducing the network overhead. When comparing both pub-sub solutions, the PubSub-Mob is the one capable of delivering more *Data* packets due to its mobility awareness feature.

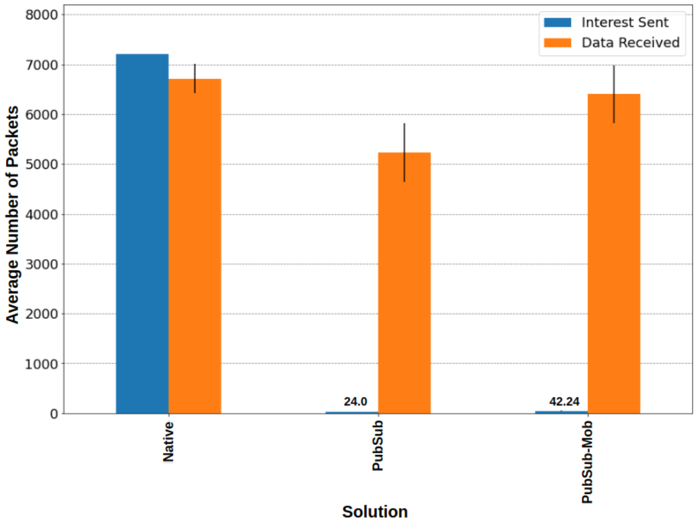


Figure 5.22: Average number of *Interest* packets transmitted and number of *Data* packets received, per mobile Consumer - scenario with real vehicular mobility traces with native, PubSub and PubSub-Mob solutions.

Table 5.27 summarizes the performance of all solutions regarding the Consumer satisfaction ratio. It shows the total number of *Interest* and *Data* packets, transmitted and received by all Consumers and the Producer. As discussed before, the native solution achieves the highest Consumer satisfaction ratio, 93.2%, due to the highest number of *Interest* packets transmitted, presenting a ratio of *Data* received per *Interest* transmitted of 0.9. Regarding the pub-sub solutions, they present a Consumer satisfaction ratio of 88.9% and 72.6%, with and without mobility awareness respectively. It should be noted that our solution offers less 4% of Consumer satisfaction ratio when compared to the Native solution, but the ratio of *Data* received per *Interest* transmitted goes from 0.9 up to 151.5.

Table 5.27: Consumer mobility content retrieval metrics - scenario with real vehicular mobility traces with native, PubSub and PubSub-Mob solutions.

Solution	Packets	Consumers	Producer	Consumer satisfaction ratio	*Data* received per *Interest* transmitted
PubSub-Mob	*Interest*	1 056	705	88.9%	151.5
	Data	160 026	29 124		
PubSub	*Interest*	600	290	72.6%	217.9
	Data	130 714	28 796		
Native	*Interest*	180 000		93.2%	0.9
	Data	167 782			

In the following, we study how the *Interest* lifetime and the *Data* transmission rate impact both pub-sub solutions. In the former we set the *Data* transmission rate to one packet every two seconds, and in the latter we set the *Interest* lifetime to 600 seconds.

Interest Lifetime

Figures 5.23 and 5.24 evaluate the impact of the *Interest Lifetime* in the performance of both pub-sub solutions.

Figure 5.23 presents the average number of subscription *Interest* packets transmitted by each mobile Consumer: as expected, as the *Interest* lifetime increases, less *Interest* packets are needed. The results also show that the number of transmitted subscription *Interests* is the same for all mobile Consumers in the PubSub solution without the mobility support (standard deviation equal to 0), because a new subscription *Interest* is only transmitted right before the expiry time in order to renew the subscription. On the other hand, in the PubSub-Mob solution, *Interest* packets have to be transmitted consequently by Expiry Events.

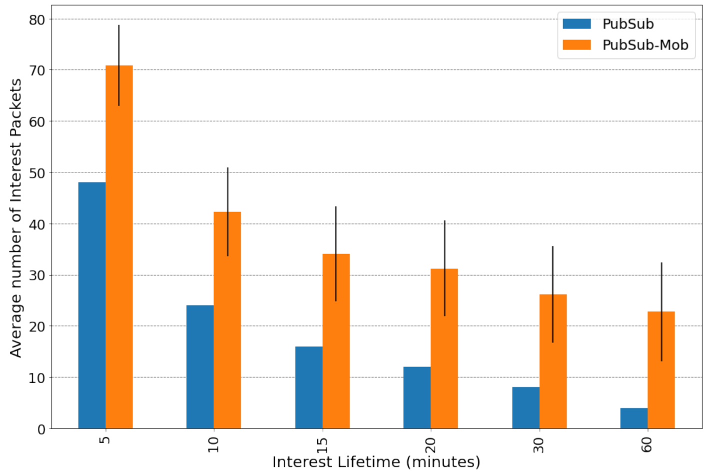


Figure 5.23: *Interest* lifetime impact in Consumer mobility using pub-sub solutions in scenarios with real vehicular mobility traces - average number of *Interest* packets transmitted, per mobile Consumer.

As for the average *Data* packets received by the mobile Consumers, illustrated in Figure 5.24, as the *Interest* lifetime increases, the pub-sub solution without mobility support fails to deliver the same amount of *Data* packets. This occurs because the reverse path to be used by the *Data* packet from the Producer up to the mobile Consumer gets outdated. Such behavior does not occur in the PubSub-Mob, due to the mobility awareness introduced by our approach.

Table 5.28 sums up the impact of the *Interest* lifetime in the PubSub-Mob solution. The first observation goes to the Consumer satisfaction ratio as the *Interest* lifetime increases: this value keeps increasing, reaching 97.4% for an *Interest* lifetime of 60 minutes. Such behavior is justified by the reuse of the reverse paths created by the SIS packets used in our solution, as the number of *Data* packets transmitted by the Producer is almost the same. Moreover, such value is always smaller when compared to the number of *Data* packets received by

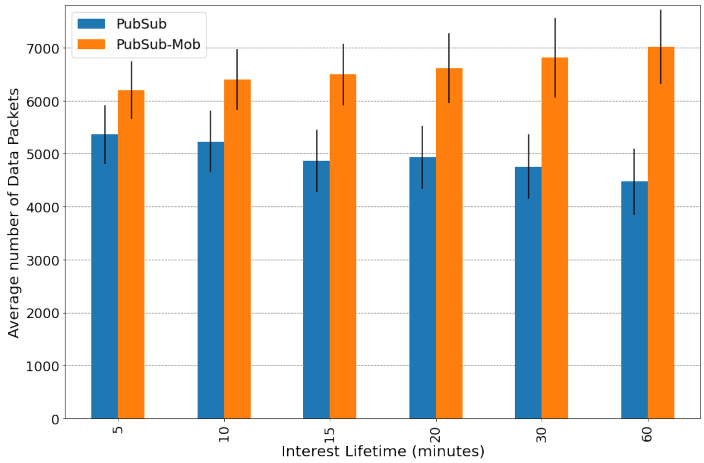


Figure 5.24: *Interest* lifetime impact in Consumer mobility using pub-sub solutions in scenarios with real vehicular mobility traces - average number of *Data* packets received, per mobile Consumer.

the Consumer, which is justified by the fact that each *Data* packet sent by the Producer is replicated in each intermediate node when several subscriptions exist on different interfaces.

Table 5.28: *Interest* lifetime impact in Consumer mobility - PubSub-Mob solution in scenarios with real vehicular mobility traces.

***Interest* lifetime [min]**	**Packets**	**Consumers**	**Producer**	**Consumer satisfaction ratio**	***Data* received per *Interest* transmitted**
5	*Interest*	1 771	997	86.1%	87.5
	Data	154 936	29 205		
10	*Interest*	1 056	705	88.9%	151.5
	Data	160 026	29 124		
15	*Interest*	852	611	90.2%	190.6
	Data	162 387	29 102		
20	*Interest*	780	552	91.9%	212.0
	Data	165 377	29 098		
30	*Interest*	654	469	94.6%	260.4
	Data	170 289	29 065		
60	*Interest*	568	407	97.4%	308.7
	Data	175 316	29 053		

Data transmission rate

Figures 5.25 and 5.26 evaluate the impact of the *Data* transmission rate in the performance of both pub-sub solutions. It is expected that, by changing this variable, the number of *Interests* transmitted by the Consumers would be the same as it does not influence the need for more *Interest* packets transmissions (Fig. 5.25). Such behavior is observed in the pub-sub solution without mobility support; however, the PubSub-Mob increases the number of *Interests* transmitted due to the configured Expiry Event. Regarding the number of *Data* packets received by the mobile Consumers (Fig. 5.26), as we decrease the *Data* transmission rate, less *Data* packets are transmitted by the Producer, and consequently, less *Data* packets are received. This results in less *Data* packets travelling the network, making the difference between both pub-sub solutions negligible.

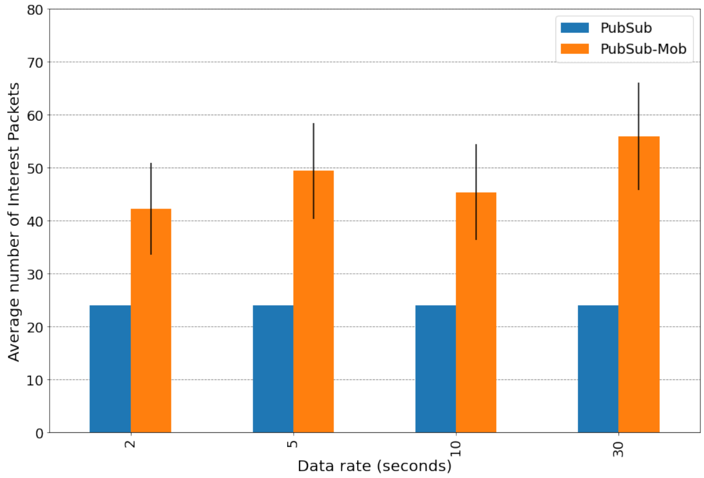


Figure 5.25: *Data* transmission rate impact in Consumer Mobility in scenarios with real vehicular mobility traces - average number of *Interest* packets transmitted, per mobile Consumer.

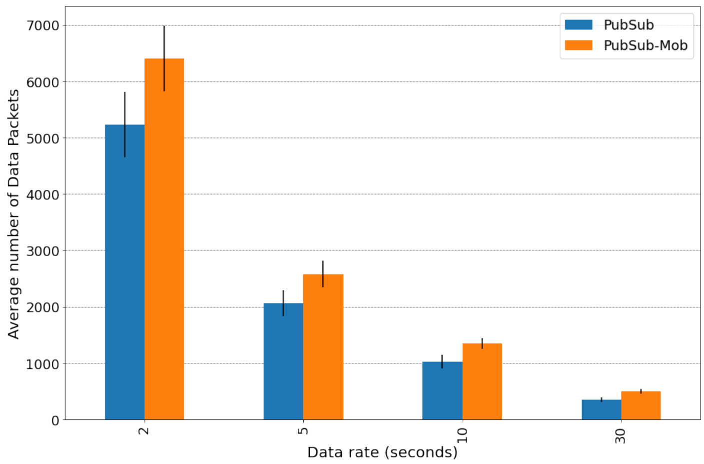


Figure 5.26: *Data* transmission rate impact in Consumer Mobility in scenarios with real vehicular mobility traces - average number of *Data* packets received, per mobile Consumer.

Table 5.29 compiles the impact of the *Data* transmission rate in the proposed pub-sub solution. The increase of the *Data* transmission rate decreases the number of *Data* packets received by the mobile Consumers, as expected. Nevertheless, with less *Data* packets travelling in the network, the Consumer satisfaction ratio increases, surpassing 100%, justified by the delivery of the same *Data* packet by different RSUs in the coverage of the same mobile Consumer. Finally, as we increase the *Data* transmission rate, the number of *Data* packets replicated by each intermediate nodes becomes smaller, decreasing the ratio of *Data* received per *Interest* transmitted, from 151.5 to 8.9.

Table 5.29: *Data* transmission rate impact in Consumer mobility - PubSub-Mob scenarios with real vehicular mobility traces.

Data **transmission rate [s]**	**Packets**	**Consumers**	**Producer**	**Consumer satisfaction ratio**	***Data*** **received per** ***Interest*** **transmitted**
2	*Interest*	1 056	705	88.9%	151.5
	Data	160 026	29 124		
5	*Interest*	1 235	700	89.6%	52.2
	Data	64 510	11 848		
10	*Interest*	1 135	740	93.6%	29.7
	Data	33 692	6 094		
30	*Interest*	1 400	768	103.5%	8.9
	Data	12 431	2 267		

5.3 Chapter considerations

This chapter focused on the evaluation of the proposed solutions targeting Consumer mobility, for both *Single-Data-Request* and pub-sub approaches, using NDN Network simulator NDNSim software.

First it was explained the two evaluation scenarios, the Functional and one with real vehicular mobility traces. Alongside, it was detailed the topology and processing stages in order to set up the different scenarios with an accurate MM.

It was also presented the involved parametrization responsible for the different conditions imposed to the network. The used metrics to evaluate the proposed solutions were also explained. Also, these metrics were used to evaluate native approaches for comparison purposes.

Overall, the obtained results clearly show that our solutions with mobility awareness outperform the native NDN communication paradigm and traditional pub-sub mechanisms, in Consumer mobile situations.

Regarding the solution focused to achieve seamless Consumer mobility in single-data request communication, for most of the evaluated occasions the proposed approach outperformed the Native approached, having a great impact on quality of service to the mobile end-users. It was also proposed a solution to alleviate the network overhead and still handle connection in Consumer mobility scenarios with mobility predictions notifications coming from the MM.

Regarding the pub-sub mobility approach, with a significant reduction of network overhead by considering only Subscription *Interest* packets and a mobility awareness mechanism to proactively fix the outdated *Data* reverse paths, the solution achieved results that outperform the traditional pub-sub mechanisms, as well as the native *Interest*-Request NDN communication paradigm, in Consumer mobility situations, reaching nearly the Consumer satisfaction ratio of the Native approach.

CHAPTER 6

Conclusions

Location-oriented networks are frequently dependant on connection-oriented protocols. Thus, for the communication to be settled, an establishment of a session between the client and server needs to be done. When client mobility is introduced, the re-establishment of these sessions is required so both parties, Consumer and Content Provider, are aware of the up-to-date network addresses, causing severe disruption and degradation of the quality of service.

ICN comes as a blueprint for the revolution of the Future Internet. A network architecture following this paradigm's principles, such as NDN, could completely change the current Internet System in terms of scalability, security and mobility, therefore satisfying the needs of future social development.

NDN and its features for content dispersion already attract attention of telecommunication industry. Nevertheless, in the case where handovers between APs happen frequently, like in vehicle communication environments, efficient mobility management has to be supported to constantly provide these services.

This dissertation proposed NDN-based approaches to cover Consumer mobility for both *Single-Data-Request* and pub-sub communications. These approaches are able to monitor and anticipate mobile nodes position trajectories and adjust to the new paths, providing seamless mobility of Consumers and improving the reliability of the network. These solutions are dependant of a network entity, acting as the MM. It is responsible to keep track of the latest information, particularly the location of the mobile Consumers, through pub-sub sessions between this entity and the RSUs scattered in the infrastructure, as well as it is responsible to deliver predictions about the *next RSUs* of Consumers to the RSU hosting them. Such delivery of predictions could be established using *Single-Data-Request* or pub-sub communication approaches. Furthermore, the RSU acts as the entity responsible of the settlement of communication path between the CP/Publisher and *next RSUs* of its mobile hosts, when potential migration of the mobile Consumers is acknowledged. For this settlement, to tackle seamless handover, it uses new support packets, IS and DS, when the used communication by the Consumer is *Single-Data-Request*, and SIS and SDS, when it is of type pub-sub. Both of these contain information about the Name of the recipients which the packet containing the content has to bifurcate. Furthermore, both DS and SDS resort to the intermediate node's RIB in order to reach the pointed recipients. The entries for the aforementioned table depend on a routing protocol based on NDN, such as NLSR, used to disseminate the topology of the RSUs, in our case. Once the content reaches the *next RSUs*, the content is cached and proactively sent to the wireless environment in the format of a *Data* packet. Moreover, if it regards pub-sub communication, the path between the *next RSU* and *Publisher* is set.

In order to implement these proposals, it was used the *ndnSIM* simulator, which implements the NDN architecture using much of the NFD source code. Dedicated applications to represent the behaviours and mechanism of RSUs, MM, Publisher and Subscribers of contents were implemented, along with the aforementioned support packets in order to establish content deviation. Furthermore, the forwarding module from the intermediate nodes had to be changed in order to handle the new packets properly, requiring modifications at the PIT module and addition of the RIB table mainly used to forward the DS and SDS packets. Moreover, modifications to handle pub-sub communications, using persistent PIT entries, were also implemented.

To evaluate the mobility awareness solutions, we have included two types of scenarios, with different network topologies: functional scenarios and scenarios using real-world vehicular mobility traces. Regarding the functional scenarios, the mobility traces are fabricated, having the freedom to establish different trajectories through time at different speeds. On the other hand, real-world vehicular mobility traces, pre-processed and filtered, were also used with the aim of verifying the performance and feasibility in a real context environment. Different simulation scenarios were set in order to evaluate the impact of different network and Consumers' parameterizations, such as: speed of the Consumers, wireless range, *Interest* frequency, link delays, number of mobile nodes, *Data* rate and *Interest* lifetime (the last two are related to pub-sub solutions). The process of setting up the scenarios also consisted on having the MM ready to deliver the *next RSUs* prediction values to achieve mobility awareness. Having in context all the mobility traces of each Consumer to be used by the simulator, it was built an MM resorting to historical dataframe to predict the *next RSUs*; for the evaluation of the pub-sub proposed approach in scenarios using real vehicular mobility traces, it was used a Markovian Model as a mobility predictor.

The results obtained in functional scenarios evaluating the *Single-Data-Request* solutions show that, in most cases, the proposed approach is able to handle Consumer mobility, outperforming the native solution, with little number of Timeouts, at the expense of an elevated network overhead cost, caused by the involved interchange of packets between the MM and RSUs. Furthermore, having high Producer delays and link delays, our solution could damage the communication because of the expected delivery time to be greater than the set Consumer Timeouts. Using the proposed solution, the content delivery time would be dependent on the RTT of the prediction request and the RTT regarding the delivery of the content. To tackle these issues, it was proposed the use of mobility prediction notifications in order to discard, from the content delivery time, the dependency of the RTT regarding the mobility prediction request. With this approach, not only we are able to obtain better results compared to the native solution, but we also diminish significantly the overhead of the network compared to the scenario where mobility prediction is asked per Consumer *Interest*. On the not-so-bright-sight, the number of timeouts per Consumer is not smaller than the aforementioned ones.

When it comes to use of real vehicular mobility traces, taking into account the built topology, the proposed solution does not find utility on the proposed solution for *Single-Data-Request*. The velocity of the mobile Consumers throughout the simulation and the dimensions of the transmission areas are enough for the retrieval of content to be established at the same RSU where the requests were sent to the ISP.

Regarding the results obtained by the proposed pub-sub solutions, for both functional scenarios and scenarios using real vehicular mobility traces, it shows that it deals seamlessly with the mobility of the Consumer, reaching high Consumer satisfaction ratio and significant reduction of Timeouts, compared to simple pub-sub solutions. As well as it achieves reduced network overhead when compared to NDN native solution.

Overall, this Dissertation was able to assess the major impacts of prediction mobility in different types of approaches, and this will be very important for the real deployment of such approaches.

6.1 Future Work

The work that was presented in this dissertation allowed the design, implementation, and evaluation of schemes for handling seamless content delivery when the Consumer mobility is introduced in a NDN-based network, with the use of a centralized network entity acting as a mobility predictor. Solutions were made for *Single-Data-Request* and pub-sub approaches.

However, this work could be improved in several ways, further improving the network overhead caused by the proposed solutions. Future work goes through the following topics:

- **Distributed Mobility Management to our architecture**
  With the advent of 5G networks, a strong increase in the scale of urban networks is expected. In order to meet this new requirement of scale and diversity of applications, we aim to include distributed mobility management to our architecture, enabling traffic to be discharged closer to the edge, making our solution more suitable for ultra-dense environments;
- **Introduction of appropriate caching mechanisms**
  Introduction of appropriate caching mechanisms, with mobility awareness, could leverage the overall performance of our solution, be it in push-based or pull-based communication models. Moreover, the development and analysis of different cache policies with the aim to persist content at the *next RSUs* until the respective interested mobile Consumer reaches its transmission area. It would reduce the network overhead if the content is ready to deliver by a nearest RSU, rather than sending an *Interest* towards the ISP.

- **Better management of pub-sub subscription paths in a mobile Consumer environment**

  When a Consumer hands-off from an AP, the subscription paths established between the AP and the Publisher might still persist and be of no use by the other Consumers at the transmission area, causing undesired network overhead. That being said, it could be relevant to study the research of policies and strategies to cancel unutilized subscription paths established by persistent PIT entries in the network.
- **Port solution to a real network testing and evaluation**

  The integration and evaluation of the proposed scheme in a testbed using NDN architecture is one of the following works. This could be done since the implementation presented by this dissertation was done using the code implemented by the NDN architecture, which already has several real-world implementations, including a universal testbed [1].
- **Implementation of fragmentation strategies for Information Centric Networks**

  The improvement of the performance of the ICN architectures by assuming fragmentation of contents, which might tackle the problem in the case of high link error rates and big size contents, decreases the chance of content delivery ratio, the expiration of *Interests*, and consequently the number of *Interests* retransmissions.

[1] https://named-data.net/ndn-testbed/

Bibliography

[1] S. Adhatarao, M. Arumaithurai, D. Kutscher, and X. Fu, "Nemoi: Network mobility in icn", in *International Conference on Communication Systems and Networks*, Springer, 2018, pp. 220–244.

[2] G. Tyson, N. Sastry, I. Rimac, R. Cuevas, and A. Mauthe, "A survey of mobility in information-centric networks: Challenges and research directions", in *Proceedings of the 1st ACM Workshop on Emerging Name-Oriented Mobile Networking Design - Architecture, Algorithms, and Applications*, ser. NoM '12, Hilton Head, South Carolina, USA: Association for Computing Machinery, 2012, pp. 1–6, ISBN: 9781450312912. DOI: 10.1145/2248361.2248363.

[3] D. Saxena, V. Raychoudhury, N. Suri, C. Becker, and J. Cao, "Named data networking: A survey", *Computer Science Review*, vol. 19, pp. 15–55, 2016, ISSN: 1574-0137. DOI: https://doi.org/10.1016/j.cosrev.2016.01.001.

[4] "Cisco visual networking index: Forecast and methodology, 2015–2020", CISCO, USA, Tech. Rep., 2016.

[5] L. Hogie, P. Bouvry, and F. Guinand, "An overview of manets simulation", *Electronic Notes in Theoretical Computer Science*, vol. 150, no. 1, pp. 81–101, 2006, Proceedings of the First International Workshop on Methods and Tools for Coordinating Concurrent, Distributed and Mobile Systems (MTCoord 2005), ISSN: 1571-0661. DOI: https://doi.org/10.1016/j.entcs.2005.12.025.

[6] L. Nassar, A. Jundi, K. Golestan, F. Sattar, F. Karray, M. Kamel, and S. Boumaiza, "Vehicular ad-hoc networks(vanets): Capabilities, challenges in context-aware processing and communication gateway", in *Autonomous and Intelligent Systems*, M. Kamel, F. Karray, and H. Hagras, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 42–49, ISBN: 978-3-642-31368-4.

[7] S. Zeadally, R. Hunt, Y.-S. Chen, A. Irwin, and A. Hassan, "Vehicular ad hoc networks (vanets): Status, results, and challenges", *Telecommunication Systems*, vol. 50, no. 4, pp. 217–241, 2012.

[8] M. Amadeo, C. Campolo, and A. Molinaro, "Information-centric networking for connected vehicles: A survey and future perspectives", *IEEE Communications Magazine*, vol. 54, no. 2, pp. 98–104, 2016. DOI: 10.1109/MCOM.2016.7402268.

[9] G. P. Pollini, "Trends in handover design", *IEEE Communications magazine*, vol. 34, no. 3, pp. 82–90, 1996.

[10] G. Miao, J. Zander, K. W. Sung, and S. B. Slimane, *Fundamentals of mobile data networks.* Cambridge University Press, 2016.

[11] C. E. Perkins, "Mobile ip", *IEEE Communications Magazine*, vol. 35, no. 5, pp. 84–99, 1997. DOI: 10.1109/35.592101.

[12] P. Nikander, A. Gurtov, and T. R. Henderson, "Host identity protocol (hip): Connectivity, mobility, multi-homing, security, and privacy over ipv4 and ipv6 networks", *IEEE Communications Surveys Tutorials*, vol. 12, no. 2, pp. 186–204, 2010. DOI: 10.1109/SURV.2010.021110.00070.

[13] G. Tyson, N. Sastry, R. Cuevas, I. Rimac, and A. Mauthe, "A survey of mobility in information-centric networks", *Commun. ACM*, vol. 56, no. 12, pp. 90–98, Dec. 2013, ISSN: 0001-0782. DOI: 10.1145/2500501.

[14] D. P. Arjunwadkar, "Introduction of ndn with comparison to current internet architecture based on tcp/ip", *International Journal of Computer Applications*, vol. 105, pp. 31–35, 2014.

[15] M. Ripeanu, "Peer-to-peer architecture case study: Gnutella network", in *Proceedings first international conference on peer-to-peer computing*, IEEE, 2001, pp. 99–100.

[16] G. Peng, "Cdn: Content distribution network", *arXiv preprint cs/0411069*, 2004.

[17] A. Passarella, "A survey on content-centric technologies for the current internet: Cdn and p2p solutions", *Computer Communications*, vol. 35, no. 1, pp. 1–32, 2012, ISSN: 0140-3664. DOI: https://doi.org/10.1016/j.comcom.2011.10.005.

[18] M. Amadeo, C. Campolo, J. Quevedo, D. Corujo, A. Molinaro, A. Iera, R. L. Aguiar, and A. V. Vasilakos, "Information-centric networking for the internet of things: Challenges and opportunities", *IEEE Network*, vol. 30, no. 2, pp. 92–100, 2016. DOI: 10.1109/MNET.2016.7437030.

[19] H. Garmani, D. Ait Omar, M. El Amrani, M. Baslam, and M. Jourhmane, "The effect of caching on cp and isp policies in information-centric networks", *Mobile Information Systems*, vol. 2020, p. 8 895 271, Oct. 2020, ISSN: 1574-017X. DOI: 10.1155/2020/8895271.

[20] D. Kutscher, S. Eum, K. Pentikousis, I. Psaras, D. Corujo, D. Saucez, T. C. Schmidt, and M. Waehlisch, "Information-centric networking (icn) research challenges", *Internet Research Task Force (IRTF)*, pp. 1–38, 2016.

[21] S. Arshad, B. Shahzaad, M. A. Azam, J. Loo, S. H. Ahmed, and S. Aslam, "Hierarchical and flat-based hybrid naming scheme in content-centric networks of things", *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1070–1080, 2018. DOI: 10.1109/JIOT.2018.2792016.

[22] M. Baugher, B. Davie, A. Narayanan, and D. Oran, "Self-verifying names for read-only named data", in *2012 Proceedings IEEE INFOCOM Workshops*, 2012, pp. 274–279. DOI: 10.1109/INFCOMW.2012.6193505.

[23] O. Ascigil, S. Reñé, G. Xylomenos, I. Psaras, and G. Pavlou, "A keyword-based icn-iot platform", in *Proceedings of the 4th ACM Conference on Information-Centric Networking*, ser. ICN '17, Berlin, Germany: Association for Computing Machinery, 2017, pp. 22–28, ISBN: 9781450351225. DOI: 10.1145/3125719.3125733.

[24] M. F. Bari, S. R. Chowdhury, R. Ahmed, R. Boutaba, and B. Mathieu, "A survey of naming and routing in information-centric networks", *IEEE Communications Magazine*, vol. 50, no. 12, pp. 44–53, 2012. DOI: 10.1109/MCOM.2012.6384450.

[25] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, "A data-oriented (and beyond) network architecture", in *Proceedings of the 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, ser. SIGCOMM '07, Kyoto, Japan: Association for Computing Machinery, 2007, pp. 181–192, ISBN: 9781595937131. DOI: 10.1145/1282380.1282402.

[26] P. Jokela, A. Zahemszky, C. Esteve Rothenberg, S. Arianfar, and P. Nikander, "Lipsin: Line speed publish/subscribe inter-networking", in *Proceedings of the ACM SIGCOMM 2009 Conference on Data Communication*, ser. SIGCOMM '09, Barcelona, Spain: Association for Computing Machinery, 2009, pp. 195–206, ISBN: 9781605585949. DOI: 10.1145/1592568.1592592.

[27] V. Jacobson, D. K. Smetters, J. D. Thornton, M. Plass, N. Briggs, and R. Braynard, "Networking named content", *Commun. ACM*, vol. 55, no. 1, pp. 117–124, Jan. 2012, ISSN: 0001-0782. DOI: 10.1145/2063176.2063204. [Online]. Available: https://doi.org/10.1145/2063176.2063204.

[28] *Named data networking: Motivation amp; details.* [Online]. Available: https://named-data.net/project/archoverview/.

[29] D. Saxena, "Named data networking: A survey", *Elsevier Computer Science Review*, Jan. 2016. DOI: 10.1016/j.cosrev.2016.01.001.

[30] *Nsf future internet architecture project*, http://www.nets-fia.net/, Accessed: 2020-09-30.

[31] J.-H. Cha, J.-H. Choi, J.-Y. Kim, Y.-H. Han, and S.-G. Min, "A mobility link service for ndn consumer mobility", *Wireless Communications and Mobile Computing*, vol. 2018, p. 5 149 724, Jul. 2018, ISSN: 1530-8669. DOI: 10.1155/2018/5149724.

[32] D. Raychaudhuri, K. Nagaraja, and A. Venkataramani, "Mobilityfirst: A robust and trustworthy mobility-centric architecture for the future internet", *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 16, no. 3, pp. 2–13, Dec. 2012, ISSN: 1559-1662. DOI: 10.1145/2412096.2412098.

[33] Y. Rao, Z. Huachun, D. Gao, H. Luo, and Y. Liu, "Proactive caching for enhancing user-side mobility support in named data networking", Jul. 2013, pp. 37–42. DOI: 10.1109/IMIS.2013.16.

[34] J. Lee and D. Kim, "Proxy-assisted content sharing using content centric networking (ccn) for resource-limited mobile consumer devices", *IEEE Transactions on Consumer Electronics*, vol. 57, no. 2, pp. 477–483, 2011. DOI: 10.1109/TCE.2011.5955182.

[35] A. V. Ventrella, G. Piro, and L. A. Grieco, "Publish-subscribe in mobile information centric networks: Modeling and performance evaluation", *Computer Networks*, vol. 127, pp. 317–339, 2017, ISSN: 1389-1286. DOI: https://doi.org/10.1016/j.comnet.2017.08.022.

[36] A. Detti, D. Tassetto, N. B. Melazzi, and F. Fedi, "Exploiting content centric networking to develop topic-based, publish–subscribe manet systems", *Ad hoc networks*, vol. 24, pp. 115–133, 2015.

[37] B. Nour, H. Ibn-Khedher, H. Moungla, H. Afifi, F. Li, K. Sharif, H. Khelifi, and M. Guizani, "Internet of things mobility over information-centric/named-data networking", *IEEE Internet Computing*, vol. 24, no. 1, pp. 14–24, 2019.

[38] A. Shariat, A. Tizghadam, and A. Leon-Garcia, "An icn-based publish-subscribe platform to deliver uav service in smart cities", in *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2016, pp. 698–703. DOI: 10.1109/INFCOMW.2016.7562167.

[39] A. V. Ventrella, L. A. Grieco, and G. Piro, "Information-centric networking in environmental monitoring: An overview on publish-subscribe implementations", in *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, IEEE, 2017, pp. 1–6.

[40] W. Drira and F. Filali, "A pub/sub extension to ndn for efficient data collection and dissemination in v2x networks", in *Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014*, IEEE, 2014, pp. 1–7.

[41] P. Fazio, F. De Rango, and M. Tropea, "Prediction and qos enhancement in new generation cellular networks with mobile hosts: A survey on different protocols and conventional/unconventional approaches", *IEEE Communications Surveys Tutorials*, vol. 19, no. 3, pp. 1822–1841, 2017. DOI: 10.1109/COMST.2017.2684778.

[42] A. Nadembega, A. Hafid, and T. Taleb, "Mobility-prediction-aware bandwidth reservation scheme for mobile networks", *IEEE Transactions on Vehicular Technology*, vol. 64, no. 6, pp. 2561–2576, 2015. DOI: 10.1109/TVT.2014.2345255.

[43] H. Zhang and L. Dai, "Mobility prediction: A survey on state-of-the-art schemes and future applications", *IEEE Access*, vol. 7, pp. 802–822, 2019. DOI: 10.1109/ACCESS.2018.2885821.

[44] Z. Zhao, L. Guardalben, M. Karimzadeh, J. Silva, T. Braun, and S. Sargento, "Mobility prediction-assisted over-the-top edge prefetching for hierarchical vanets", *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 8, pp. 1786–1801, 2018. DOI: 10.1109/JSAC.2018.2844681.

[45] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, k. claffy kc, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, "Named data networking", *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, pp. 66–73, Jul. 2014, ISSN: 0146-4833. DOI: 10.1145/2656877.2656887.

[46] J. Shi and B. Zhang, "NDNLP: A Link Protocol for NDN", The University of Arizona, Jul. 2012.

[47] L. D. J. Leira, *Caching baseado no contexto em ICN para ambientes IoT móveis*. [Online]. Available: http://hdl.handle.net/10773/29537.

[48] A. Compagno, M. Conti, C. Ghali, and G. Tsudik, "To nack or not to nack? negative acknowledgments in information-centric networking", pp. 1–10, 2015.

[49] B. Nour, K. Sharif, F. Li, S. Yang, H. Moungla, and Y. Wang, "ICN Publisher-Subscriber Models: Challenges and Group-based Communication", *IEEE Network*, vol. 33, no. 6, pp. 156–163, 2019. DOI: 10.1109/MNET.2019.1800551.

[50] J. Zhang, Q. Li, and E. M. Schooler, "iHEMS: An information-centric approach to secure home energy management", in *2012 IEEE Third International Conference on Smart Grid Communications (SmartGridComm)*, IEEE, Nov. 2012. DOI: 10.1109/smartgridcomm.2012.6485986.

[51] L. Wang, V. Lehman, A. Hoque, B. Zhang, Y. Yu, and L. Zhang, "A secure link state routing protocol for ndn", *IEEE Access*, vol. PP, pp. 1–1, Jan. 2018. DOI: 10.1109/ACCESS.2017.2789330.

[52] K. Mershad, H. Artail, and M. Gerla, "Roamer: Roadside units as message routers in vanets", *Ad Hoc Networks*, vol. 10, no. 3, pp. 479–496, 2012, ISSN: 1570-8705. DOI: https://doi.org/10.1016/j.adhoc.2011.09.001.

[53] H. Zhang and L. Dai, "Mobility Prediction: A Survey on State-of-the-Art Schemes and Future Applications", *IEEE Access*, vol. 7, pp. 802–822, 2019. DOI: 10.1109/ACCESS.2018.2885821.

[54] S. Mastorakis, A. Afanasyev, and L. Zhang, "On the evolution of ndnsim: An open-source simulator for ndn experimentation", *SIGCOMM Comput. Commun. Rev.*, vol. 47, no. 3, pp. 19–33, Sep. 2017, ISSN: 0146-4833. DOI: 10.1145/3138808.3138812.

[55] S. Mastorakis, A. Afanasyev, I. Moiseenko, and L. Zhang, "Ndnsim 2.0: A new version of the ndn simulator for ns-3", Jan. 2015.

[56] T. R. Henderson, M. Lacage, G. F. Riley, C. Dowell, and J. Kopena, "Network simulations with the ns-3 simulator", *SIGCOMM demonstration*, vol. 14, no. 14, p. 527, 2008.

[57] M. Amadeo, C. Campolo, and A. Molinaro, "Forwarding strategies in named data wireless ad hoc networks: Design and evaluation", *Journal of Network and Computer Applications*, vol. 50, pp. 148–158, 2015.

[58] *Named data networking forwarding daemon (nfd) 0.7.1 documentation.* [Online]. Available: https://named-data.net/doc/NFD/current/.

[59] *Ndn-cxx: Ndn c++ library with experimental extensions 0.7.1 documentation.* [Online]. Available: https://named-data.net/doc/ndn-cxx/current/INSTALL.html.

[60] S. Mastorakis, A. Afanasyev, and L. Zhang, "On the evolution of ndnsim: An open-source simulator for ndn experimentation", *ACM SIGCOMM Computer Communication Review*, vol. 47, no. 3, pp. 19–33, 2017.

[61] P. M. Santos, J. G. P. Rodrigues, S. B. Cruz, T. Lourenço, P. M. d'Orey, Y. Luis, C. Rocha, S. Sousa, S. Crisóstomo, C. Queirós, S. Sargento, A. Aguiar, and J. Barros, "PortoLivingLab: An IoT-Based Sensing Platform for Smart Cities", *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 523–532, 2018. DOI: 10.1109/JIOT.2018.2791522.

[62] L. Irio, A. Ip, R. Oliveira, and M. Luís, "An adaptive learning-based approach for vehicle mobility prediction", *IEEE Access*, vol. 9, pp. 13 671–13 682, 2021. DOI: 10.1109/ACCESS.2021.3052071.