UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

MATHEUS BRENNER LEHMANN

PDRM: A Proactive Data Replication Mechanism to Improve Content Mobility Support in NDN using Location Awareness

Thesis presented in partial fulfillment
of the requirements for the degree of
Doctor of Computer Science

Advisor: Prof. Dr. Marinho Pilla Barcellos

Porto Alegre
November 2017

CIP — CATALOGING-IN-PUBLICATION

Lehmann, Matheus Brenner

PDRM: A Proactive Data Replication Mechanism to Improve Content Mobility Support in NDN using Location Awareness / Matheus Brenner Lehmann. – Porto Alegre: PPGC da UFRGS, 2017.

118 f.: il.

Thesis (Ph.D.) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2017. Advisor: Marinho Pilla Barcellos.

1. Named Data Networking. 2. Content Mobility. 3. Data Replication. 4. Location Awareness. I. Barcellos, Marinho Pilla. II. Título.

“All that is gold does not glitter,
Not all those who wander are lost;
The old that is strong does not wither,
Deep roots are not reached by the frost.”
— J. R. R. TOLKIEN

ACKNOWLEDGMENTS

I would like to acknowledge and thank the following people for helping me during my Ph.D. course.

- Paulo and Suzana Lehmann. My parents and the rest of my family for their continuous support of my decisions.
- Marinho Barcellos. My advisor for guiding my research learning process since 2008 when I was an undergraduate student.
- Andreas Mauthe. My unofficial co-advisor for his valuable contribution to the development of my Ph.D. thesis.
- Arun Venkataramani, Jörg Ott, Antonio “Guto” Rocha, Luciano Gaspary, Miguel Campista, and Weverton Cordeiro. The committee of my thesis proposal and thesis defenses for their thoughtful comments that helped improve my final thesis work.
- Rodolfo Antunes, Rodrigo Mansilha, and Lucas Leal. My ICN-colleagues for the research collaboration on this amazing topic throughout my Ph.D. course.
- Flavio “Barata” Santos, Weverton Cordeiro, and Lucas Muller. My colleagues for being a positive influence on my academical and personal growth.
- Rodrigo Tambara, Gabriel “Careca” Hernandez, and Daniela Quinsani. My closest friends for keeping me sane with beer, DotA, and other games.
- Many other people that somehow participated in my journey.

ABSTRACT

The problem of handling user mobility has been around since mobile devices became capable of handling multimedia content and is still one of the most relevant challenges in networking. The conventional Internet architecture is inadequate in dealing with an ever-growing number of mobile devices that are both consuming and producing content. Named Data Networking (NDN) is a network architecture that can potentially overcome this mobility challenge. It supports *consumer mobility* by design but fails to offer the same level of support for *content mobility*. Content mobility requires guaranteeing that consumers manage to find and retrieve desired content even when the corresponding producer (or primary host) is not available. In this thesis, we propose PDRM, a Proactive and locality-aware Data Replication Mechanism that increases content availability through data redundancy in the context of the NDN architecture. It explores available resources from end-users in the vicinity to improve content availability even in the case of producer mobility. Throughout the thesis, we discuss the design of PDRM, evaluate the impact of the number of available providers in the vicinity and in-network cache capacity on its operation, and compare its performance to Vanilla NDN and two state-of-the-art proposals. The evaluation indicates that PDRM improves content mobility support due to using object popularity information and spare resources in the vicinity to help the proactive replication. Results show that PDRM can reduce the download times up to 53.55%, producer load up to 71.6%, inter-domain traffic up to 46.5%, and generated overhead up to 25% compared to Vanilla NDN and other evaluated mechanisms.

Keywords: Named Data Networking. Content Mobility. Data Replication. Location Awareness.

PDRM: Um Mecanismo Proativo de Replicação de Dados para Melhorar o Suporte à Mobilidade de Conteúdo em NDN usando Consciência de Localização

RESUMO

O problema de lidar com a mobilidade dos usuários existe desde que os dispositivos móveis se tornaram capazes de lidar com conteúdo multimídia e ainda é um dos desafios mais relevantes na área de redes de computadores. A arquitetura de Internet convencional é inadequada em lidar com um número cada vez maior de dispositivos móveis que estão tanto consumindo quanto produzindo conteúdo. *Named Data Networking* (NDN) é uma arquitetura de rede que pode potencialmente superar este desafio de mobilidade. Ela suporta a *mobilidade do consumidor* nativamente, mas não oferece o mesmo nível de suporte para a *mobilidade de conteúdo*. A mobilidade de conteúdo exige garantir que os consumidores consigam encontrar e recuperar o conteúdo desejado mesmo quando o produtor correspondente (ou o hospedeiro principal) não estiver disponível. Nesta tese, propomos o PDRM (*Proactive Data Replication Mechanism*), um mecanismo de replicação de dados proativo e consciente de localização, que aumenta a disponibilidade de conteúdo através da redundância de dados no contexto da arquitetura NDN. Ele explora os recursos disponíveis dos usuários finais na vizinhança para melhorar a disponibilidade de conteúdo, mesmo no caso da mobilidade do produtor.w Ao longo da tese, discutimos o projeto do PDRM, avaliamos o impacto do número de provedores disponíveis na vizinhança e a capacidade de cache na rede em sua operação e comparamos seu desempenho com NDN padrão e duas propostas do estado-da-arte. A avaliação indica que o PDRM melhora o suporte à mobilidade de conteúdo devido ao uso de informações de popularidade dos objetos e recursos extras na vizinhança para ajudar a replicação pró-ativa. Os resultados mostram que o PDRM pode reduzir os tempos de download até 53,55%, o carregamento do produtor até 71,6%, o tráfego entre domínios até 46,5% e a sobrecarga gerada até 25% em comparação com NDN padrão e os demais mecanismos avaliados.

Palavras-chave: Named Data Networking, Mobilidade de Conteúdo, Replicação de Dados, Consciência de Localização.

LIST OF ABBREVIATIONS AND ACRONYMS

NDN	Named Data Networking
PDRM	Proactive Data Replication Mechanism
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
ICN	Information-Centric Network
CCN	Content-Centric Networking
IRTF	Internet Research Task Force
CDN	Content Delivery Network
P2P	Peer-to-Peer
FIB	Forwarding Information Base
CS	Content Store
PIT	Pending Interest Table
LCE	Leave a Copy Everywhere
LCD	Leave a Copy Down
VANET	Vehicular Ad Hoc Network
QoE	Quality of Experience
AS	Autonomous System
IoT	Internet of Things
MP	Mobile Producer
LRU	Least Recently Used

LIST OF FIGURES

LIST OF TABLES

CONTENTS

1 INTRODUCTION

Mobile traffic has been increasing year on year and represents a considerable share of network traffic. At the end of 2016, there were 8 billion mobile devices that generated 7.2 exabytes of traffic per month (CISCO, 2017). These values represent an increase of almost half a billion (429 million) devices and 63% of monthly traffic compared to the previous year (CISCO, 2017). Even with this growing trend, today's Internet architecture fails to provide adequate user mobility support despite mechanisms such as Mobile IPv4 (PERKINS, 2010) and Mobile IPv6 (JOHNSON; ARKKO; PERKINS, 2015). The cause is a mismatch between the Internet architecture and the continually evolving user demands.

Recently, new proposals have been discussed to change the Internet architecture significantly. One of the most promising proposals to meet the user demands, including support for user mobility, is the Information-Centric Networking (ICN) (AHLGREN et al., 2012), a network paradigm that shifts from a host-centric approach (current Internet) to a content-centric one. The most prominent network architecture proposal in the ICN model is the Named Data Networking (NDN)[1] (JACOBSON et al., 2009).

NDN is a network architecture proposed to address existing shortcomings of the current Internet, including mobility. Inspired by the predominant user behavior (i.e., the interest of users in content rather than sources or delivery mechanisms), NDN focuses on content, instead of hosts, and builds the network architecture around it. The development of NDN has gained considerable traction with support from industry (CAROFIGLIO, 2017; POLAKOS, 2016; ITU, 2017; NSF/INTEL, 2016) and its contribution to the work of the IRTF ICN Research Group[2]. Overall, NDN has shown to be a promising solution to handle the growing trend of content dissemination and mobility (CISCO, 2017) as well as to be the core component of mobility support in the 5G research and development (ANDREWS et al., 2014).

The NDN architecture is designed to help the producers to disseminate and keep their content available. Three features stand out to support content mobility: content in NDN is replicable across the network, in-network caching can increase the number of content copies, and any device holding a content copy can satisfy incoming requests for it.

---

[1] The Content-Centric Networking (CCN) (MOSKO et al., 2015) is a similar network architecture proposal that shares many principles with NDN. In this thesis, we discuss NDN, but the work could be applied to CCN with minimal changes. For more information, refer to <http://www.ccnx.org/> or <http://named-data.net/project/faq/#How_does_NDN_differ_from_Content-Centric_Networking_CCN>.

[2] <https://irtf.org/icnrg>

As a consequence, content can have additional *providers* besides the producer (or primary host) for its dissemination.

On the one hand, the content-centric design allows the NDN architecture to provide native support for consumer mobility. The communication model employed is receiver-driven and connectionless, which enables consumers to resume content retrieval after moving by just resending their requests seamlessly. This support is simpler and more efficient than Mobile IPv6 (JOHNSON; ARKKO; PERKINS, 2015). On the other, the NDN design is not sufficient to address *content mobility*, offering only limited support and requiring more complex network solutions to achieve this goal (TYSON et al., 2013; KUTSCHER et al., 2016; ZHANG et al., 2016).

1.1 Hypothesis

The challenge of content mobility compared to consumer mobility is that objects have to be kept available and reachable for consumers despite a possible movement or unavailability of its providers (ANASTASIADES; BRAUN; SIRIS, 2014). With the possibility of multiple providers for the same content in NDN, content is available if there is at least one content provider with a copy of it in the network. Thus, NDN ensures content mobility support when consumers can continue to retrieve some content despite the unavailability of a fraction of its providers (producer included) due to movement. Given the problem stated, the following question guides this thesis.

How to improve content mobility support in NDN using an approach that will not only benefit from but also enhance NDN's content-centric design principles?

1.2 Thesis Contributions

In this thesis, we investigate how to use proactive and location-aware data replication to improve the content mobility support in NDN. We identify the proactive content replication as an underexplored approach for content mobility support in NDN and study ways to implement it while benefiting the most from the NDN architecture. As a result, we propose PDRM (Proactive Data Replication Mechanism), composed of two operations: Vicinity Discovery and Content Push.

The goal of PDRM is to increase content availability through data redundancy with efficient use of network and end-user resources. PDRM is an optional service for mobile producers to replicate content proactively. Unlike previous approaches, a mobile producer using PDRM learns about its vicinity and uses this information to influence the replication decisions. The learning step is the key feature that allows PDRM to optimize the content replication process for high dissemination performance and low resource consumption.

The design of PDRM addresses the content mobility support as follows. The proactive replication enables new object copies to be created and disseminated before consumers request the content, which enhances the use of NDN features in a subsequent content distribution. Being aware of the location context allows the content producer to collect information about its objects (e.g., popularity) and other users in the vicinity. This knowledge can be used to infer which objects are prone to become popular and if end-users are willing to become temporary providers of such content. The extra storage in end-users helps the network opportunistically to increase content availability and improve its dissemination.

We evaluate PDRM performance and overhead under various aspects. For the evaluation, we implement PDRM and state-of-the-art proposals in the ndnSIM (MASTORAKIS et al., 2015). First, we measure the impact of the number of available providers and in network cache capacity on the performance of PDRM. Then, we compare PDRM to Vanilla NDN and state-of-the-art approaches, namely Data Spot (WOO et al., 2014) and Data Depot (JACOBSON et al., 2012), in the face of producer unavailability periods.

Experiments indicate that, overall, PDRM improves content mobility support in many scenarios, particularly those with limited cache capacity and low producer availability. The proactive data replication executed by PDRM increases content availability and, consequently, reduces the consumer download time. PDRM outperforms Vanilla NDN and the evaluated mechanisms, benefiting the network in a myriad of ways. PDRM reduces all evaluated metrics, achieving improvement as high as the following: the download times by 53.55%, producer load by 71.60%, inter-domain traffic by 46.50%, and generated overhead by 25%. These benefits are more evident in the cases of low producer availability than of high availability.

The main contributions of the thesis are summarized as follows.

1. A detailed investigation of proactive and location-aware data replication to support content mobility in NDN.
2. The proposal of PDRM (Proactive Replication Data Mechanism), whose main differential is learning about the vicinity to improve the content mobility support in

NDN.

3. The evaluation of PDRM performance and overhead under different scenario configurations and comparative to Vanilla NDN and two state-of-the-art proposals (Data Depot and Data Spot).

1.3 Organization

The rest of this thesis is organized as follows.

Chapter 2 presents the background context of this thesis. First, we define the content mobility challenge and the approaches to address it. Then, we describe the NDN architecture regarding network elements, communication model, and user mobility support.

Chapter 3 discusses the state-of-the-art proposals in NDN content mobility that focus on improving content availability through data redundancy.

Chapter 4 describes PDRM. First, we present an overview of PDRM. Then, we detail the design and implementation of the two main operations: Vicinity Discovery and Content Push. Finally, we discuss the design decisions, showing how they differ from the state-of-the-art.

Chapter 5 describes the methodology employed in the evaluation of PDRM. The evaluation comprises an investigation of the impact of the number of available providers and in-network cache capacity on its performance, and a comparison of PDRM to Vanilla NDN and two state-of-the-art proposals. We discuss the extensions made to ndnSIM, scenario configurations, and metrics.

Chapter 6 presents the results obtained from the evaluation of PDRM. We analyze the results of PDRM operation regarding performance and overhead in a total of four scenarios. The first two focuses solely on PDRM performance while the last two compares PDRM to state-of-the-art proposals.

Chapter 7 concludes this thesis with a summary of the main insights and future work ideas.

2 BACKGROUND

In this chapter, we present the background context of this thesis in two sections: content mobility and Named Data Networking (NDN). The first section defines content, its mobility challenge, and the possible approaches for a network architecture to handle it. The second section describes the NDN architecture regarding its elements, communication model, and mobility support.

2.1 Content Mobility

Content is the fundamental concept of the NDN architecture, which represents it as individual pieces of data called objects. NDN identifies objects with globally unique and hierarchical names and defines them as the basic transfer unit for the communication between users through Interest and Data packets. When transferring objects, NDN divides them into fixed-size chunks and send them as idempotent Data packets[1]. The Data packets also contain an embedded cryptographic signature, enabling them to be stored and reused in future requests by any device in the network. With these designs choices, chunks, and consequently objects, can be replicable and cacheable in the network because any element with a copy of them has the information needed to guarantee data authenticity and integrity.

The mobility of content depends on the status of existing copies in the network and the devices holding them. Initially, there is only the original content copy generated by the producer (i.e., the original provider). During the content lifetime, events may change the number of content copies and their location in the network. Content availability increases with the addition of content copies in three situations: consumers retrieve objects, providers reconnect to the network, and routers cache passing packets. Conversely, content availability decreases with the removal of content copies due to users deleting stored objects, providers disconnecting from the network, and routers evicting cached packets.

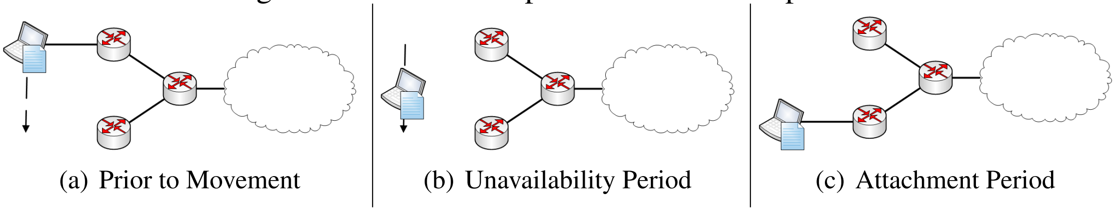
The content mobility challenge is to keep content available (i.e., to have at least one reachable copy of it), minimizing the negative impact of events that reduce content availability. A set of factors may affect positively or negatively the availability and dy-

---

[1] Idempotence describes an operation or element that produces the same result whenever it is executed. In this case, it refers to NDN Data packets, which can satisfy any incoming request from any consumer for a given object in the network.

namics of the copies. They are subject to the placement and replacement policies used in the caches and the mobility behavior of end-users. The content popularity may impact the cached content, tending to benefit popular objects in detriment of unpopular ones. The passing traffic together with the router caching policies may result in more stable or volatile cached copies. Lastly, the user churn impacts the reliability of copies found on end-users.

Figure 2.1: Movement periods of a content provider



(a) Prior to Movement (b) Unavailability Period (c) Attachment Period

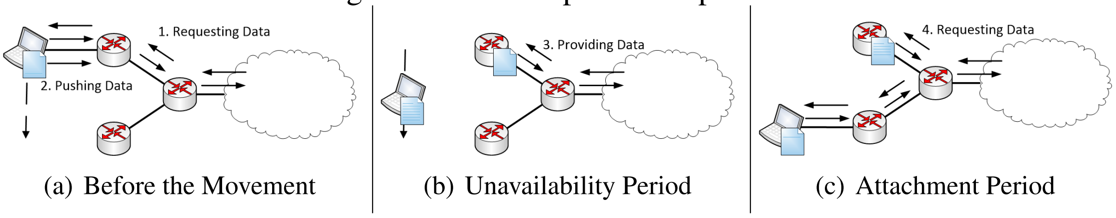
We define the movement of a content provider with two distinct periods: unavailability and (re)attachment. The former is characterized by the provider's potential lack of network connectivity during movement while the latter refers to the provider (re)joining the network and re(establishing) connectivity. Figure 2.1 exemplifies the movement process of a provider. First, the provider is connected to the network at some location and decides to move somewhere else (Figure 2.1(a)). Then, during the movement, it is disconnected and unavailable to serve data (Figure 2.1(b)). Finally, the provider reaches its new location and restores its connectivity (Figure 2.1(c)), enabling it to provide its content again.

In this thesis, we focus on the unavailability period, which occurs when a provider is unable to serve its content. The primary challenge that arises from it is keeping the consumer session continuity despite the potential adverse effects of mobility on the content availability (ANASTASIADES; BRAUN; SIRIS, 2014). A network architecture can address the unavailable provider with three approaches: Data Replication, Communication Restoration, and Provider Tracking.

Data Replication

The first approach uses data redundancy and caching to keep the content available despite the provider mobility. These features enable other elements besides the original provider to serve the requested data, allowing content to be available even when its primary provider is not. This kind of approach is commonly used in content dissemination, such as Content Delivery Network (CDN) (SAROIU et al., 2002), P2P file sharing (e.g., BitTorrent (COHEN, 2003)), and ICN (AHLGREN et al., 2012). Particularly about ICN and NDN, data redundancy is well suited to their principles because it focuses on the content instead of the hosts.

Figure 2.2: Data replication operation



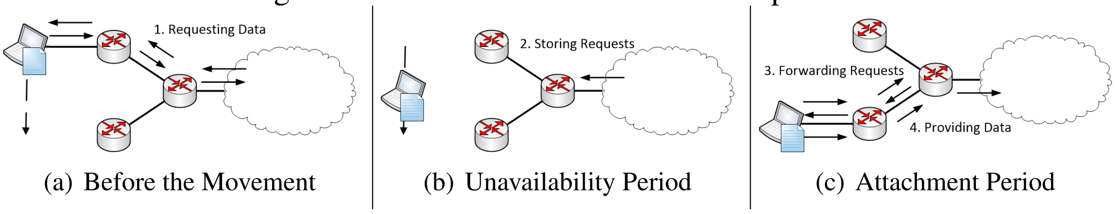
(a) Before the Movement (b) Unavailability Period (c) Attachment Period

Figure 2.2 exemplifies the use of data replication to handle provider mobility. Before the movement, the provider serves the requested data and also pushes its content to the network (Figure 2.2(a)). During the unavailability period, shown in Figure 2.2(b), the router that received the pushed content can respond to incoming requests on behalf of the moving provider, maintaining the content available. In the attachment period, the provider updates its location and consumers can retrieve the content from either source in the network, as seen in Figure 2.2(c).

Communication Restoration

The communication restoration is the second approach and aims at restoring the communication between users without loss of messages. A device stores the incoming Interest packets destined for the unavailable provider. Once the provider rejoins the network, it fetches the stored messages from the storing device and resumes the communication. This approach has a high focus on the original content producers but could be adapted for each device to manage a set of providers for the same content, reducing the dependency of a particular host that provides the requested object.

Figure 2.3: Communication restoration operation



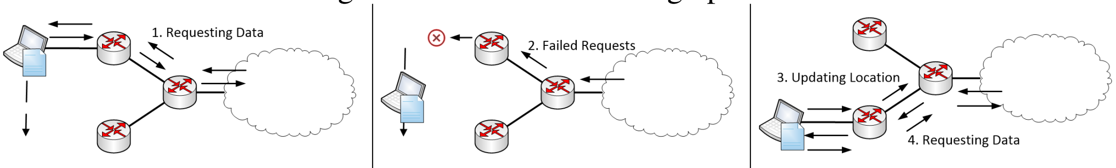
(a) Before the Movement (b) Unavailability Period (c) Attachment Period

Figure 2.3 shows an example of this operation. Before the movement, the provider serves the requested data as usual (Figure 2.3(a)). During the unavailability period, shown in Figure 2.3(b), the edge router identifies no routable path for the provider and stores the incoming requests destined for the moving provider. In the attachment period, the provider contacts the router, to fetch the stored requests, and provides the data for them, as seen in Figure 2.3(c). The communication resumes normally after that.

Provider Tracking

The third approach focuses on reducing the provider unavailability period, especially during hand-off, by monitoring the link state or predicting movement. The primary goal is a coordination between the provider and network to maintain seamless mobility during the movement. The seamless support quality depends on the provider being mobility-aware (i.e., noticing its movement and notifying the network) and the network having a quick and efficient hand-off process. Similar to the communication restoration, this approach also has a high focus on a particular host than on the content itself. It could be adapted to consider a set of providers and locate any available when a mobility event occurs.

Figure 2.4: Provider tracking operation



(a) Before the Movement (b) Unavailability Period (c) Attachment Period

Figure 2.4 demonstrates the provider tracking operation. Before the movement, the provider serves the requested data normally (Figure 2.4(a)). During the unavailability period, shown in Figure 2.4(b), the requests fail because the network does not know the current location of the provider. Alternatively, the network could track the provider or another source during the movement, achieving seamless content mobility during the entire process. In the attachment period, the provider updates its location, enabling it to receive and respond to consumer requests again, as seen in Figure 2.4(c).

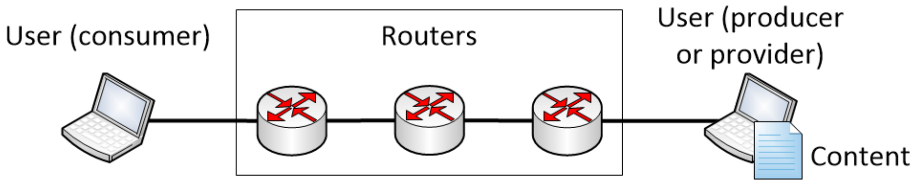
2.2 Named Data Networking

This section describes the Named Data Networking (NDN), a network architecture proposal based on the ICN paradigm. NDN shifts the network architecture's focus from a host-centric perspective to a content-centric one. It promotes content to be the central element of the network through the use of objects as the basic data representation. The NDN communication is receiver-driven: consumers send Interest packets for objects and providers respond with Data packets containing the required content. This design choice allows the architecture to focus on what the consumer wants instead of where it should request the data. Throughout the section, we discuss the elements of the network, the communication model along with its operation, and the current mobility support of NDN.

Network Elements

There are three central components of an NDN network: content, users, and routers. Figure 2.5 exemplifies them in a simple NDN network. In the example, there are three routers that route and forward messages and two users connected to them. The user on the left acts as a consumer that wants a piece of content while the one on the right is the producer or a provider of the given content. Next, we describe users and routers in the NDN network, and how they interact with content.

Figure 2.5: NDN network components



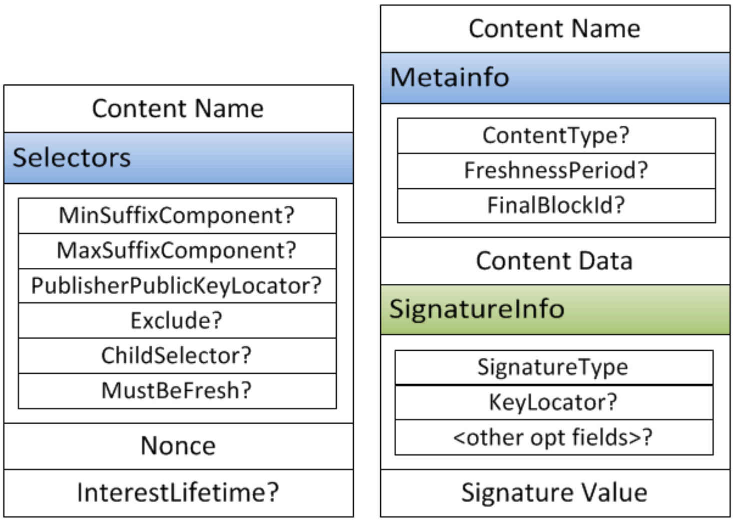
In the mobility context of this thesis, **users** are mobile entities and move according to a movement pattern. They act as consumers (retrieving objects), producers (generating content) or providers (serving content objects). Users are available when connected to the network, enabling them to ask for or provide objects. They can become unavailable (for instance, due to movement or downtime), a period in which they cannot receive or transmit data, making their stored content unavailable to other consumers. Lastly, every user associates to a device, which has resource restraints, such as storage space, bandwidth, and battery lifetime.

Routers are static devices of the network that neither consume nor produce content. Instead, they route and forward Interest and Data packets in the network. The routing process can be done using either the Forwarding Information Base (FIB) table or schemes (e.g., broadcast) that do not rely on such routing information. Despite not actively requesting content, routers can cache objects opportunistically and use them to serve future requests of interested consumers. However, due to the passing traffic, the content in their caches can be very volatile, offering no guarantee of availability.

Communication Model

The NDN architecture has a receiver-driven and connectionless communication model. The communication is exchanged based on two kinds of packets shown in Figure 2.6: Interest and Data. The Interest packet (Figure 2.6(a)) has two required fields: the content name and the nonce. The content name identifies each object uniquely in the network while the nonce is a randomly generated string that combined with the name should uniquely identify Interest packets and avoid routing loops. The selectors are optional filters that consumers use to restrict further the data that providers may send in response to Interest requests. For instance, the consumer can specify a particular publisher using the "PublishPublicKeyLocator" field. At last, the Interest packet also has an optional lifetime field, which defines how long that Interest will be valid before it times out.

Figure 2.6: NDN basic packet types

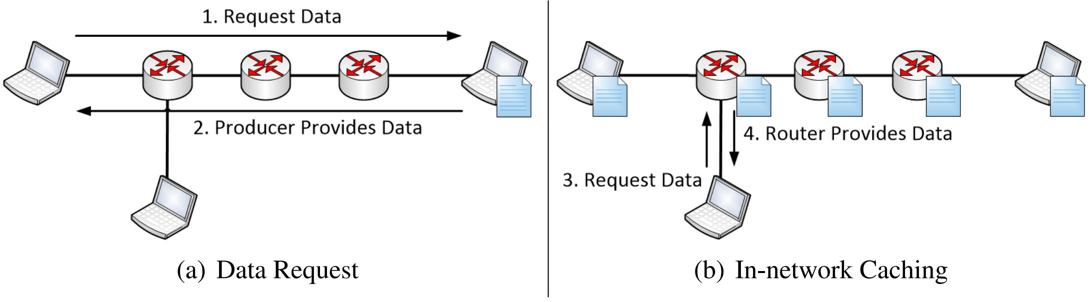


(a) Interest Packet (b) Data Packet

The Data packet, shown in Figure 2.6(b), represents an object in the network. The required fields of this packet are the content name, data, signature, and signing information. The content name and data are straightforward: the former uniquely identifies the object while the latter is the object itself. The signature is the result of signing the Data packet and is used to guarantee its authenticity. The signature information field describes the signing method employed in the Data packet. That is, the signature algorithm and, optionally, the key location if the consumer needs to retrieve it. The Data packet also has optional metainfo fields, which add extra information describing the object. For example, it can specify the packet type and for how long that object is valid (freshness).

To retrieve a content object, consumers send an Interest packet with its name. Providers, on their turn, respond to it by sending a Data packet with the requested object. Any element in the network that has a copy of the content can act as a provider opportunistically, including the routers due to the in-network caching feature. Figure 2.7 presents a simple example of two successive retrievals for the same content object. First, the consumer on the left sends an Interest packet for a given object. Routers route the packet to its provider using the FIB information, and the provider responds with the object data (Figure 2.7(a)). When the routers forward the data to the consumer, they may cache it to satisfy future requests. Then, in Figure 2.7(b), the consumer on the bottom requests the same content object. Instead of the router routing the Interest packet towards the provider, it responds the request with the previously cached data.

Figure 2.7: NDN communication model



(a) Data Request

(b) In-network Caching

The NDN router uses three data structures and at least one communication face[2] for routing and forwarding packets: Content Store (CS), Pending Interest Table (PIT), and Forwarding Information Base (FIB). They enable not only the routing and forwarding but also in-network caching and request aggregation. The CS is a cache that stores content objects temporarily. The PIT records all forwarded Interest requests whose response has neither been received nor expired yet. The FIB is the routing table, which manages the configured routes to names on the network. Besides these data structures, the router also has faces to communicate with other applications on the same host or devices in the network.

When an NDN router receives an Interest packet, it executes the following actions. First, it verifies if the requested object is in the CS. If that is the case, the router sends a Data packet, responding the request. Otherwise, it checks the PIT if there is already

---

[2] An interface is named face in NDN

an ongoing request for the given object. If such entry is in the PIT, the router aggregates the requests and does not forward the current. In case there is no pending Interest for that object, the router looks in the FIB where it should route the Interest packet, adds a PIT entry for the object, and sends the packet in the selected face. Finally, if none of the previous conditions are met, the router can discard the packet, broadcast it, send a NACK back, or execute any other action.

The routing process is performed only for the Interest packets, whereas the Data ones come back through the reverse path. When a router processes an Interest packet, it leaves a breadcrumb in the PIT, marking where it should forward the response. With the PIT information, routers can aggregate requests, forwarding multiple Data packets back to requesters using a single PIT entry. If an NDN router receives a Data packet of an object that has no pending requests, it is marked as unsolicited data and discarded.

Figure 2.8: NDN router organization and operation

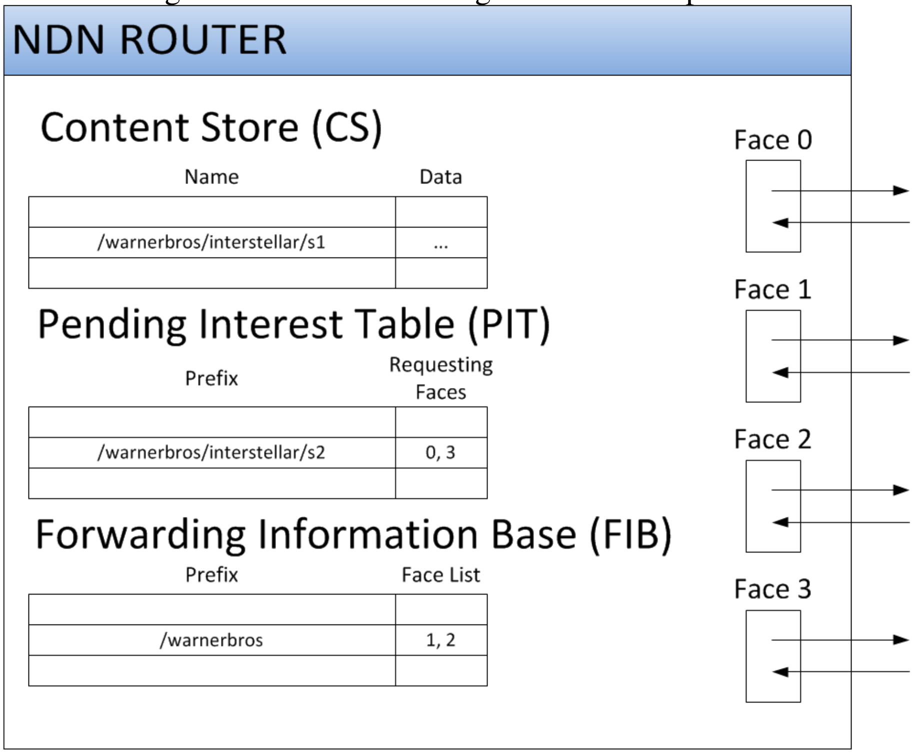


Figure 2.8 shows the organization and operation of the NDN router. In this example, the router has four external faces and the three default data structures: CS, PIT, and FIB. It also shows the operation during the retrieval of the movie “Interstellar.” The CS indicates that the first chunk (denoted s1 or segment 1) has already been retrieved by consumers and is currently cached, enabling the router to respond to any future request for it. The PIT shows that consumers from faces 0 and 3 have requested the second chunk and are currently waiting for the data. When the router receives the data for this object,

it will store a copy in the CS and forward the Data packet to all requesting faces (0 and 3 in the example). If the router receives a request for the third chunk of the movie, there will not be any information in the CS or PIT. Thus, the router checks its FIB for route information of the given object and decides to which face send the interest request. In this case, it could route the Interest packet to face 1, face 2, or both.

Mobility Support

The NDN architecture natively supports the consumer mobility due to the receiver-driven and connectionless communication model. It enables consumers to resume their object requests without needing to restore a session or connection or update their locations. Hence, consumers only need to re-send their Interest requests to continue the download without any reconfiguration. Additionally, the packet idempotence eases the communication resume by making every object independent of session or connection contexts.

For *content mobility* support, the NDN architecture provides the following features: packet idempotence and in-network caching. In this context, the packet idempotence allows different providers of the same content to send the data transparently to the consumer (similar to BitTorrent). The in-network caching is a data replication feature that increases the number of object copies in the network. However, it alone proves to be insufficient (TYSON et al., 2013; KUTSCHER et al., 2016; ZHANG et al., 2016) mainly because of its reactive operation. Since caching operates based on incoming requests, mobility hampers the initial dissemination of content that is unavailable and consumers cannot find on the network. Given this limitation, we claim that NDN requires additional mechanisms to support content mobility adequately.

3 STATE-OF-THE-ART

In this thesis, we investigate content mobility support in NDN focused on improving data availability through redundancy, known as data rendezvous approach (ZHANG et al., 2016). There are three strategies to increase the number of object copies in the network (and consequently, its availability): Data Caching, Data Spot, and Data Depot. The first one is a reactive, cache-based process according to the observed traffic whereas the last two strategies are based on replication, which is a proactive operation executed independently (and possibly prior) to consumer requests. Table 3.1 summarizes the existing proposals and guides the discussion throughout this chapter.

Proactive replication is currently an underexplored approach in NDN, particularly when addressing content mobility. Previous work suggests that proactive replication hardly helps content dissemination and could be replaced by edge caching combined with a simple replacement scheme (SHARMA; VENKATARAMANI; SITARAMAN, 2013; FAYAZBAKHSH et al., 2013). However, unlike previous work, we focus on content mobility in ICN, which alters the ratio between replication cost and miss penalty due to content unavailability. Instead of a simple cache miss, consumers may fail to find and obtain an object from the network. Therefore, a proactive replication approach can be effective under these conditions and worth the investigation.

Table 3.1: Summary of the State-of-Art in NDN Content Mobility using Data Redundancy

Reference	Method	Strategy
(LAOUTARIS; CHE; STAVRAKAKIS, 2006) (PSARAS; CHAI; PAVLOU, 2012) (SAHA; LUKYANENKO; YLä-JääSKI, 2013) (WANG; ZHANG; BENSAOU, 2013) (MICK; TOURANI; MISRA, 2016)	Caching	Data Caching
(WOO et al., 2014) (KO et al., 2014) (VASILAKOS et al., 2012) (MAJEED et al., 2017) (GRASSI et al., 2014) (SILVA; CAMPISTA; COSTA, 2016) (MAURI et al., 2017)	Replication	Data Spot
(JACOBSON et al., 2012)	Replication	Data Depot

3.1 Data Caching

The *Data Caching* strategy is native in the NDN architecture. Routers cache Data responses that they forward back to consumers and use them to satisfy future requests.

The routers decide which Data packets to store based on caching policies, such as Leave Copy Everywhere (LCE), Leave Copy Down (LCD) (LAOUTARIS; CHE; STAVRAKAKIS, 2006), and Probabilistic caching (PSARAS; CHAI; PAVLOU, 2012). These policies are simple and improve content dissemination (particularly of popular content) without incurring significant processing overhead to routers. The downside of caching regarding mobility is its reactive nature, as it caches content already available. If a content object is unavailable, it is not served to consumers and, consequently, not cached by routers.

The routers may also employ more sophisticated caching schemes with a degree of cooperation and collaboration among them (SAHA; LUKYANENKO; YLä-JääSKI, 2013; WANG; ZHANG; BENSAOU, 2013; MICK; TOURANI; MISRA, 2016). Their overall goal is to reduce the overlapping content in the network caches. Consequently, they improve the overall caching performance by increasing the number of unique objects cached. Although these sophisticated caching schemes provide better performance than simpler ones, they also add higher processing overhead on routers and do not change the reactive nature of caching, which is detrimental when producers are unavailable, and content has not spread in the network yet.

3.2 Data Spot

The *Data Spot* strategy is the most popular replication approach in NDN because it fits well in the architecture principles by taking advantage of locality to disseminate content. It associates produced content to particular areas in the network where the content should be more interesting and requested, usually around the location of the source or the majority of consumers. For example, regional content tends to have a more substantial number of consumers inside of its area of interest and a significantly smaller potential consumer outside it. Using the Data Spot strategy, the content would only be replicated inside the given region because creating copies elsewhere would be non-optimal and potentially a waste of resources.

Authors in (WOO et al., 2014; KO et al., 2014) propose Data Spot based mechanisms that allow producers to push content to the access router proactively. The proposals are based on two observations: requests are usually routed to the content source location (especially if its availability is uncertain), and the request sequence is predictable because consumers request object chunks sequentially. The solutions monitor the link state to detect when the connection strength weakens, indicating an imminent movement. Before moving, producers offload the content data of current incoming requests to the access router as unsolicited data (i.e., there is no prior request for this data). The access routers are extended to cache instead of discard the unsolicited data, keeping the content available near its source location to satisfy future requests on behalf of the content producer. The downside of these proposals is adding unsolicited data packets, which introduce new vulnerabilities to the architecture and do not guarantee the content availability due to cache volatility.

The proposals presented in (VASILAKOS et al., 2012; MAJEED et al., 2017) focus on replicating data and pushing it closer to the consumers. The Selective Neighbor Caching (VASILAKOS et al., 2012) proactively replicates data responses in a subset of proxies in the neighborhood to which the mobile consumers may move next. The decision is based on delay, cache cost, and mobility pattern of users. The proposal discussed in (MAJEED et al., 2017) focuses on pre-caching chunks of large content objects (e.g., videos) before the consumer requests them. Given the request predictability of such content objects and the overall cache capacity, the network cooperates to minimize the content distance to the consumers and the number of replicas created. The use of routers for proactive content caching used by the proposals can be troublesome for three reasons: it increases their complexity, which limits their packet processing; they have limited cache storage (less than end-users); their cache is volatile and requires changes to store objects for a longer period.

The Data Spot strategy is particularly prevalent in ad-hoc networks (e.g., VANETs). The following three papers propose mechanisms in such environments (GRASSI et al., 2014; SILVA; CAMPISTA; COSTA, 2016; MAURI et al., 2017). Authors in (GRASSI et al., 2014) proposes a vehicular network based on NDN that disseminates content using a broadcast medium. This characteristic enables any device close to the sender to store a copy of the transmitted content data, which results in content replication around the source location. The devices storing content can satisfy incoming requests, rebroadcast it to spread further or physically move themselves and the content in the network. TraC (SILVA; CAMPISTA; COSTA, 2016) is a proposal that, similar to (VASILAKOS et al., 2012), attempts to proactive replicate data in the future direction of consumers to improve data delivery in VANETs. It proposes two forwarding strategies and a neighborhood discovery protocol to push data to access points in the consumer trajectories toward their destination. The last one, presented in (MAURI et al., 2017), proposes an optimization solution to distribute content in VANETs with infrastructure accounting for cache and link bandwidth capacity. The idea is to prefetch content objects at static network nodes, enabling consumers to retrieve them faster from different access points as they move in the network. The proposals based on ad-hoc networks are hardly generalizable because they are usually domain-restricted and tailored to the target environment. That is, they take advantage of particular characteristics not found in traditional networks, such as the broadcast medium and routing performed by end-users.

3.3 Data Depot

The *Data Depot* strategy replicates the produced content in fixed servers independent of the source location that serve the received objects on behalf of the producer. In other words, it uses the available resources in data centers instead of those found in users nearby the source location as the Data Spot. This strategy is widely deployed currently in CDN and cloud storage solutions but has been underexplored in the context of NDN. If successfully implemented, Data Depot has the potential of incorporating those existing application-layer solutions to NDN's network layer.

One of the few data depot proposals in NDN is a custodian-based solution for content sharing, which takes advantage of the producer's resources across multiple devices (JACOBSON et al., 2012). After producing a piece of content, the producer can send it to a trusted device that will become its custodian and may serve it in future requests. The routers are extended to keep a mapping from prefix to custodian to endpoint, creating an overlay of the user's device and allowing them to find the best content copy according to the producer preferences (e.g., prioritize devices with continuous power and connectivity). The custodian mechanism has poor performance when a given content object is unavailable when requested. It will search a copy in all configured devices, and if it fails to find the object, it will leave a request to fetch as soon as a device with the object comes online again. In this case, the custodian mechanism has the same performance and limitations as the NDN in-network caching.

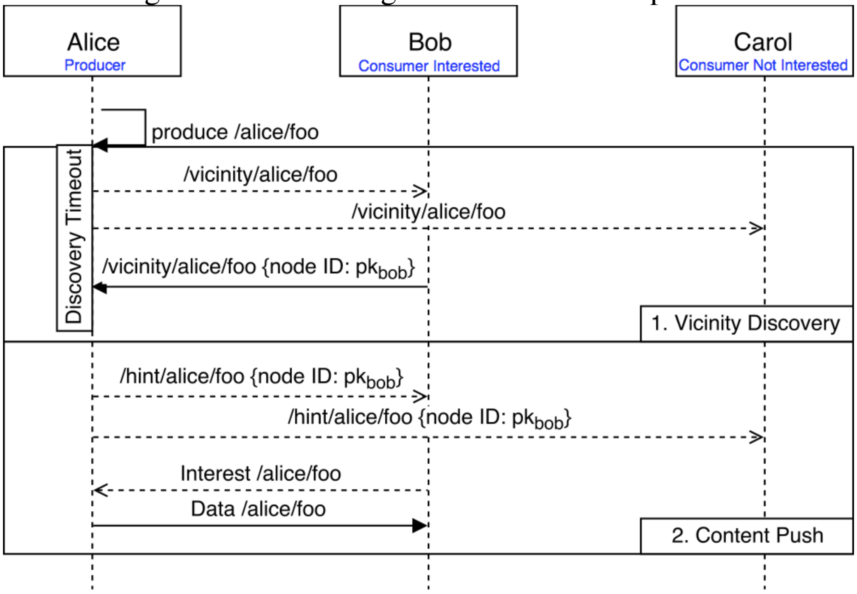
4 PDRM

In this chapter, we discuss the details of PDRM (Proactive Data Replication Mechanism), our proposal to improve the content mobility support in NDN. This version of PDRM is an enhancement from the one described in (LEHMANN; BARCELLOS; MAUTHE, 2016) and takes into account the insights obtained from its evaluation. PDRM reduces the impact of producer unavailability on the availability of content through data redundancy, with efficient use of available network and user resources. After a producer creates a content object, it can push copies to other users, which become providers. The expected results are an increase in the content retrieval rate and, consequently, an improvement to the consumer's QoE, particularly its download time, with limited overhead for users and the network.

4.1 Overview

PDRM is a proactive, locality-aware, best-effort, and hint-based replication mechanism that explores available resources from end-users in the vicinity. It is composed of two operations: Vicinity Discovery and Content Push. To proactively deal with mobility, a producer may execute them on the objects of choice one or more times after their creation. First, the producer sends a probe to learn about surrounding devices and the popularity of content in them. Then, the producer decides based on the collected information whether to replicate the objects of choice using the content push operation.

Figure 4.1: Time diagram of the PDRM operation



The execution of PDRM is illustrated with an example composed of Alice (producer), Bob and Carol (users in the vicinity of Alice). Figure 4.1 presents the message exchange of PDRM's execution in this example. It gives an overview of the execution flow of PDRM whereas the details of both Vicinity Discovery and Content Push operations are discussed in the following sections.

The example begins with Alice creating the object named </alice/foo> and triggering a vicinity discovery for it. Alice broadcasts a probe to its vicinity, which contains Bob and Carol. We assume Bob is interested in the content, but Carol is not. Both receive the probing message and act accordingly to their interests. Bob replies to it with his public key while Carol just drops the message.

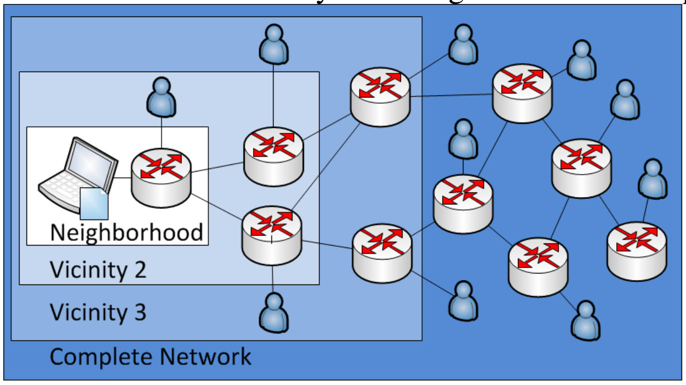
After some period to collect the responses, Alice concludes the vicinity discovery and uses the gathered knowledge to replicate her produced object. In this example, Alice checks that only Bob is interested in becoming a provider for </alice/foo> and sends him a hint for the content using a broadcast message with his public key. Similar to the vicinity probe, they both receive the hint message and act accordingly. Bob proceeds to request the object using the default NDN protocol. Carol drops the received hint because it is not destined to her.

4.2 Vicinity Discovery

The proactive replication of popular content towards potential consumers improves its availability and dissemination performance. Thus, the first step of PDRM is to obtain relevant context information about the producer location that might help the replication. Mainly, the producer learns about existing devices in the vicinity and the content popularity in them. The collected information indicates the objects most likely to be requested and their potential consumers. Using this approach, PDRM allocates the limited resources available in a mobile environment to replicate those objects that potentially will gain the most benefits.

The vicinity of a producer is the set of nodes within a distance range that can vary from the direct neighbors to the complete network (e.g., a domain or AS). Figure 4.2 exemplifies the different vicinity levels a producer may have. The inner square represents the knowledge a producer has in Vanilla NDN: only the router directly connected to it (referred as neighborhood or vicinity one). PDRM expands the size of the vicinity, allowing a producer to know more about the network. In the example, there are two intermediate vicinity levels represented by size two and three. This expansion enables a producer to learn about users (notice that each user figure accounts for a set of users) connected to one and three routers respectively. Lastly, in an extreme case, the producer can know about every user and device on the domain network, depicted by the outer square.

Figure 4.2: Different levels of vicinity from neighborhood to complete network



To discover the vicinity, the producer broadcasts an Interest probe message within a limited scope defined by the vicinity size. As the response, the producer receives Data messages from all users in the vicinity interested in the given content and willing to provide it on behalf of the producer. If the producer receives no response, it will not replicate the object due to the lack of interest for the content in the vicinity.

The evaluation of the vicinity size impact on the PDRM performance performed in (LEHMANN; BARCELLOS; MAUTHE, 2016) shows that the smallest vicinity that contains potential mobile providers is the best choice. It enables the producer to replicate its objects, increasing the content hit rate and decreasing download time, while limiting the overhead generated because it restricts the communication close to the producer. If the producer fails or wishes to find potential mobile providers in the vicinity, it can expand the vicinity size to increase the amount of discovered devices.

During the vicinity discovery, PDRM may request a set of information from the devices in the vicinity. The minimum subset is comprised of the consumer identifier and interest in the content (i.e., is the user a future consumer?). Additionally, PDRM may also collect the device availability percentage, its home network location, or any other relevant information. Even before disseminating a given object, there are at least two ways that producers could identify future consumers. The first one is using metadata to describe the content, enabling users to set their content preferences and match with the produced objects. The second is to use a subscribing scheme (e.g., a social network), which would indicate users that consume content from specific producers.

In this thesis, we employ the term *mobile provider* for generality. That does not exclude potential providers that may be fixed somewhere in the network. One promising example is domains or ASes installing a server in their border router to act as a potential provider for PDRM. In this case, the user would represent the aggregation of users in the domain, and its interest would be a result of the most significant interests found in the domain given the most requested content in it. With this approach, domains could act as CDN nodes and prefetch relevant content, potentially improving their consumer performance and also reducing traffic cost.

All the information gathered by the producer during the vicinity discovery is used to influence the object replication and copies placement. Because of the unreliability of the data provided by other users, the producer keeps the collected knowledge as soft state. If there is a scenario in which the information needs to be more reliable, PDRM could be extended in at least two non-exclusive ways. First, it could implement an accountability module to track and verify information exchanged between users, similar to those used in private BitTorrent systems. Second, the producer could periodically monitor the vicinity and build historical information about it.

We implement PDRM' vicinity discovery operation in NDN by extending the NDN Interest packet and routers to enable TTL[1], strategy selectors, and long-lived PIT entries as explained next. To trigger the vicinity discovery, the producer broadcasts an Interest packet with the extended fields TTL and strategy selectors to the reserved namespace </vicinity/<object_name>>. The TTL field contains the packet's maximum number of hops, restricting the scope of propagation to the configured vicinity. The strategy selectors, on their turn, define which information the producer is requesting from the potential providers of the content, as discussed previously.

We extend the routers to process the reserved namespace </vicinity> differently from regular Interest packets. These probe packets create a long-lived PIT entry that is removed only after its expiration period, allowing the requester to receive multiple responses. In comparison, routers consume PIT entries of regular Interest packets after they receive an object that satisfies the request. This modification extends the lifetime of PIT entries slightly (i.e., few seconds at most), which should not add new exploits and abuses to the PIT data structure (GASTI et al., 2013; VIRGILIO; MARCHETTO; SISTO, 2013; WäHLISCH; SCHMIDT; VAHLENKAMP, 2013).

---

[1] NDN does not use TTL currently, but the CCN architecture employs it with the purpose of avoiding routing loops (MOSKO et al., 2015).

4.3 Content Push

After the producer builds a view of the consumer interest in its vicinity based on the collected information during the discovery period, PDRM decides how many copies to push (replication degree) and whom to push (placement policy). Because NDN does not have a native pushing primitive, PDRM adds a sender-driven communication operation based on a hinting scheme to enable the producer to push its content proactively to other devices. Next, we discuss the design and implementation of these three aspects: replication degree, placement policy, and push communication.

The increase in content availability obtained by PDRM is related to the replication degree (i.e., the number of copies pushed). The naive approach is to imagine that every object can have multiple copies, making them always available in the network. However, this approach is unrealistic for three reasons. First, proactively pushing content can be expensive, especially for producers with limited resources. Second, the more copies are created, the less effective they become (LEHMANN; BARCELLOS; MAUTHE, 2016). Third, spare resources from end-users are finite and shared between all producers.

The evaluation of the replication degree impact on PDRM performance presented in (LEHMANN; BARCELLOS; MAUTHE, 2016) corroborates the intuition that the naive approach is inviable. The analysis concludes that pushing a single copy is enough to achieve most of the benefits from proactive content replication and a second extra copy improves further the content mobility support. However, pushing three or more copies becomes expensive for the producer while only increasing the results marginally.

Therefore, we expect small replication degrees to be affordable and, even so, not applicable to all objects and producers. PDRM defines the actual number of copies pushed based on the measured content popularity obtained during the vicinity discovery. This design aims at maximizing the benefits of data replication and reducing the waste of resources of a costly pushing operation by allocating more resources to objects that have higher popularity and are more likely to be requested in the future.

Despite NDN being highly effective in disseminating and reactively replicating trendy objects through its in-network caching, we still focus on proactively replicating the top popular objects. With this decision, PDRM aims to achieve the best individual cost-benefit ratios for data replication. Consequently, PDRM gives an early start to the dissemination of the selected content, increasing its future availability despite the potential producer unavailability. In the future, we may avoid replicating a percentage of the most popular objects and target those objects that are popular but do not benefit much from NDN caching (i.e., second tier popular objects).

After deciding how many copies to create, PDRM selects which users in the vicinity will receive a copy, enabling them to provide the content. PDRM can use any placement policy to rank users in the vicinity and select a provider based on the collected information during the vicinity discovery. For example, a producer could select the most reliable device regarding availability or the one that has the most resources to spare for replication.

However, results evaluating different placement policies presented in (LEHMANN; BARCELLOS; MAUTHE, 2016) indicate that picking a random interested user in the vicinity is good enough. This policy tends to distribute the load among the set of existing users and avoids overloading a subset of users considered to be the best providers. Further, obtaining reliable information regarding other users in a mobile environment is very costly and very difficult to use in the medium and long-term for behavior prediction.

Independent on the placement policy, one fundamental aspect to take into consideration is the willingness of users to become content providers on behalf of others. The user motivation to provide a particular content object depends on the data disseminated, the resources currently available in the device, its altruism degree, and existing incentives. In particular, incentives are notably important to increase user participation due to the limited resources that mobile devices have.

We list three possible reasons that users could lend their resources to disseminate other people's content. The first reason, as discussed in the previous section, is domains setting up a PDRM user to represent the aggregate interest in their networks. It serves two purposes: improve content dissemination and reduce traffic cost. The second reason is based on user voluntarism to disseminate content. This case is seen nowadays in social networks or instant messaging, in which people forward content (e.g., texts, images, videos) they found or received. The last reason considers a reward system. One could imagine selling the resources temporarily for others (i.e., similar to current cloud systems) or applying reciprocity to enable users to join the system (e.g., BitTorrent).

Lastly, we discuss the sender-driven communication operation based on a hinting scheme added by PDRM to NDN. It is paramount that such extension does not conflict with NDN's design principles, especially its receiver-driven model. In PDRM, the producer suggests a given content object to other users. A user that receives the indication may follow it or not. In the case of using the hint, the user requests the content and retrieves it from the producer using the NDN protocol. When replicating the object, the producer sends it to the future provider, populating the caches of routers in the path according to their caching policies.

PDRM implements the hinting scheme at the application layer using the NDN receiver-driven communication primitives and complying with NDN design philosophy. Because the producer does not always know a routable name that reaches the selected user, it broadcasts an Interest packet encrypted with the targeted user public key received during the vicinity discovery to the reserved namespace </hint/<object_name>>. We extend the routers so that the reserved namespace </hint> does not leave a trail in the PIT because it does not expect a Data object in response. Instead, upon receiving a hint, if the targeted user accepts the suggestion, it retrieves the hinted object by sending Interest packets normally. Otherwise, it justs discards the hint like any other user that receives the hint and cannot decrypt it.

4.4 Design Discussion

We develop PDRM as a retro-compatible mechanism to the NDN architecture. It means that PDRM benefits fully from any feature as well as possible extensions present in NDN. For instance, any security or access control solution can be applied to PDRM with little to no changes. The development of PDRM took into account the limitations found in the state-of-the-art proposals to guide its design.

As a result, PDRM has the following characteristics that differentiate it from other state-of-the-art proposals: *proactivity*, *locality-awareness*, *best-effort*, *hint-based*, and *flexibility*. The proactivity and locality-awareness are the most relevant features that stand PDRM out from other state-of-the-art proposals. Being best-effort, hint-based, and flexible make PDRM a well-round and generalizable mechanism that complies with the NDN design philosophy. Next, we discuss each of these characteristics.

The *proactive* approach decouples the replication from the consumer requests, enabling producers to decide when to create copies. With this design, producers can replicate content before they move even if it was not requested, making content available during the mobility period. Consequently, PDRM overcomes and complements the reactive nature of caching, which works well with content that is available or already spread but fails to deal appropriately with mobile content that is not disseminated yet.

Being *aware of locality* enables PDRM to discover spare resources from other users and use them efficiently. During the vicinity discovery, PDRM gathers information about existing devices nearby and their interest in content objects (i.e., content popularity), helping decide which of them to replicate. The leverage of nearby resources and information aims at maximizing the performance gain regarding content availability and download time while consuming the least amount of resources.

The content replication is done in a *best-effort* fashion and does not require accurate information from the users in the vicinity. Consequently, producers can replicate content under any network condition such as high mobility, dynamicity, or unreliability. On the downside, PDRM does not provide any guarantees regarding its operation.

The *hinting scheme* used by PDRM to push content follows the receiver-driven NDN communication model. It adds a single Interest packet per object at the beginning of the transmission to suggest an object to another user. Afterward, the content retrieval is done using the default NDN protocol. Compared to other approaches that alter the NDN architecture and routers significantly, the hinting scheme avoids adding vulnerability issues or processing overhead and still fully benefits from the architectural features (e.g., caching, routing).

Lastly, PDRM is *flexible* for producers to use it in various application domains. In scenarios where the information and resources are costly or unreliable (e.g., IoT and VANET), PDRM can replicate content using limited information of the vicinity and few resources. In conditions of trusted knowledge and abundant resources (e.g., CDN or cloud-storage applications), PDRM can take advantage of them to provide better content mobility support. Hence, PDRM could benefit a broader set of users if incorporated to NDN as a native feature.

5 METHODOLOGY

In this chapter, we present the methodology employed in the PDRM evaluation. We investigate the impact of the number of available providers and in network cache capacity on PDRM as well as compare PDRM to Vanilla NDN and state-of-the-art proposals for content mobility support. The evaluation was performed by extending the widely popular ndnSIM simulator (MASTORAKIS et al., 2015). Throughout the chapter, we detail the extensions made to ndnSIM, the scenarios evaluated, and the metrics measured to perform the PDRM evaluation.

5.1 ndnSIM

ndnSIM (MASTORAKIS et al., 2015) is the most complete and realistic NDN simulator available, enabling every aspect of NDN to be simulated. We extended ndnSIM with the implementation of the evaluation model, which comprises a Mobile Producer (MP) application that uses only the NDN default content mobility support. This case is denoted as *Vanilla NDN* and is the basis for the implementation of PDRM and the two mechanisms based on state-of-the-art proposals: *Data Depot* and *Data Spot*. Next, we discuss the extensions added to ndnSIM[1].

Evaluation Model

The evaluation model describes an NDN network with user mobility and serves as the basis to study PDRM. The fundamental elements of the model are the devices and the content objects in the network. A device is characterized by a mobility pattern, interest in content objects, and available resources. Devices can be either users or routers. User devices are mobile elements that produce, provide, and consume content. Routers are stationary components that form the network infrastructure, route requests, forward data, and cache content objects to serve future requests.

A content object is a piece of information with some popularity that is produced and consumed by users. Objects are composed of one or more chunks and may have different sizes. A producer **creates** a content object and can execute three actions after-

---

[1] The source code used in the evaluation is available at <https://github.com/mblehmann/ndnSIM>

ward: notify the availability of the object, push it to other devices, and satisfy Interest requests for it. The notification action announces the created object and the location of the producer to the network, making it reachable for consumer requests. Also, every time a content provider moves and wants to continue providing objects, it needs to notify the network about its new location to update the routing information. The announcement information is propagated to every router in the network to converge their routing tables.

After creating a content object, the producer can push it to other devices using PDRM, which decides how many replicas are generates and where to place them on the network. Each device receiving a pushed copy of the object announces its possession. It results in the device becoming a provider of the object and being able to satisfy any future Interest requests for it. The routers in the path between the producer and a new provider may store a copy of it according to their caching policy and update their routing information for this content object with the new provider.

Whether an object was pushed or not, it may be sought by a consumer **request**. When users request objects, they retrieve data from the nearest provider according to the routing information in routers, which is built based on announcements. If a router in the path between the consumer and the nearest provider has the object in its cache, it will provide the data for the consumer instead of the provider. During the retrieval, the content object is cached in the routers in the path between the provider and the consumer according to the caching policy used in the network. If not a single provider is available during the request, the retrieval fails, and the consumer re-issues a new Interest request later.

Throughout the lifetime of a piece of content, copies of the object may be **cached** by both users and routers. Users can retrieve any content object they desire but only become providers of the subset of objects that they decide to announce. Routers use the NDN's default caching policy Leave a Copy Everywhere (LCE) that caches all data that goes through them. The cache of every device has a maximum size, which forces the device to replace old entries when caching an object in an already full cache. The caches use the Least Recently Used (LRU) policy to substitute cached objects.

Mobile Producer

The MP performs three periodic actions: start a session, move, and publish a new object. The first action stops a moving producer and initiates a session at the current location, an active period during which the producer is connected and can provide its content. The second one is the reverse: the producer stops the current session and begins moving. During a movement, the producer is disconnected from the network and does not satisfy consumer requests. The third action produces a new content object and makes it available for interested consumers to request it.

Vanilla NDN

The Vanilla MP relies solely on NDN native in-network caching to support content mobility, as described in Chapter 2.

PDRM

The MP with the implementation of the proposed mechanism PDRM, as described in Chapter 4. Besides the Vanilla MP actions, it proactively replicates some of its content objects in the vicinity with the goal of increasing the number of object copies and, consequently, their availability. The implementation of PDRM extends the Interest packet and routers to enable producers to collect information about the vicinity as well as proposes a hint-based content push operation to replicate content.

Data Depot

The Data Depot mechanism is implemented based on the custodian proposal (JACOBSON et al., 2012). Its operation is similar to PDRM: the producer pushes the created objects to a selected device in the network, making it the content custodian. The difference between using Data Depot and PDRM is the learning process of potential providers (or custodians). Data Depot has an a priori trusted device at some arbitrary location (e.g., its home network) to which all content is pushed to, similar to a CDN or cloud storage service. In contrast, PDRM discovers on-demand nearby nodes (in the vicinity) to serve as providers. Hence, the Data Depot mechanism is implemented as a sub-case of PDRM in which producers do not execute the vicinity discovery operation but instead have a previously configured trusted device.

Data Spot

The Data Spot mechanism is based on the proposal that uses unsolicited data (WOO et al., 2014). Like PDRM, it pushes content in the network to satisfy future requests. However, Data Spot differs in the content pushing and selection, as explained next. The mechanism forces content to the access router, instead of using hints to send it to other users. For that matter, it extends the Data packet with an unsolicited data flag that allows it to be sent and cached by routers rather than discarded. Concerning content selection, the mechanism decides what objects to push based on the content popularity observed from the incoming requests. To obtain this behavior, we extend the MP application to offload the objects expected to be requested during the unavailability period, just before moving.

5.2 Scenario Configuration

We design scenarios to understand the impact of the number of available providers and in-network cache capacity on the performance of PDRM as well as to compare PDRM with state-of-the-art proposals for content mobility. In all scenarios, there is a single producer[2], which follows an on/off model: it becomes unavailable for some non-negligible period, longer than real-time movement. There are multiple consumers which, in their turn, continuously request content that is periodically generated by the producer. Next, we discuss the choice of parameters common to all scenarios, summarized in Table 5.1. The scenario-specific ones will be presented before the discussion of the respective results in the following section.

Figure 5.1 illustrates the network topology used in the evaluation. The routers form a complete binary tree with height 2, representing a content dissemination tree within a domain, similarly to (FAYAZBAKHSH et al., 2013). The producer is connected to the access router, represented by the tree root, and two consumers are connected to each leaf node. Each router in the network can cache up to 1% of the total catalog size (BAYHAN et al., 2016). All links have the same capacity, with 30ms delay and 10Gbps bandwidth (GARCIA-LUNA-ACEVES; MIRZAZAD-BARIJOUGH; HEMMATI, 2016).

The producer uses one of the content mobility support mechanisms to distribute

---

[2] Note that focusing on a single producer does not affect the generality of results as seen by the similar results obtained using one and multiple producers in (LEHMANN; BARCELLOS; MAUTHE, 2016).

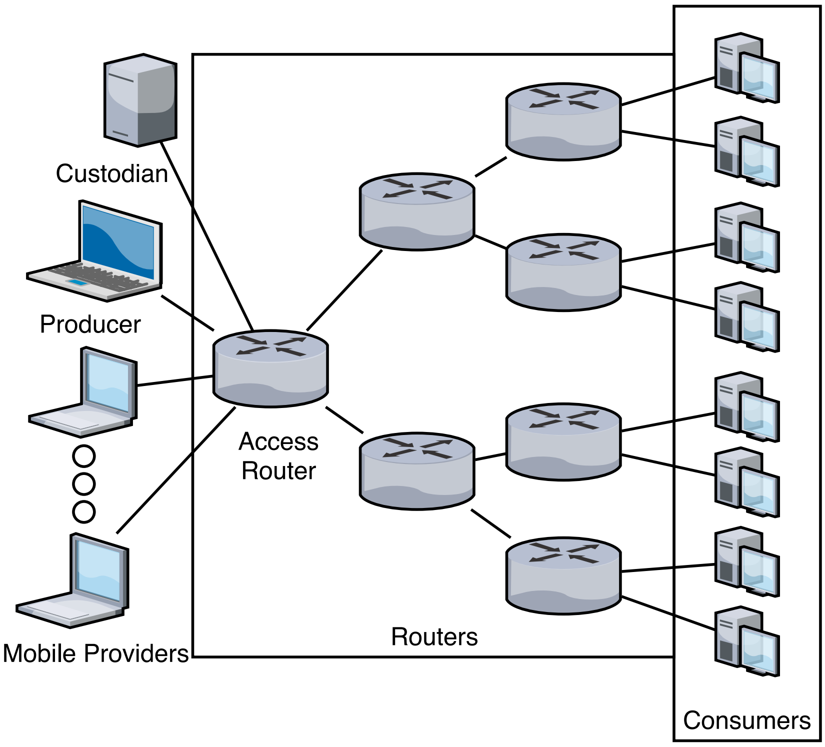
Table 5.1: Summary of the Parameters Common to All Scenarios

Parameter	Value
Topology	Complete Binary Tree (FAYAZBAKHSH et al., 2013)
Tree Height	2 (FAYAZBAKHSH et al., 2013)
Routers	15
Producers	1
Consumers	8
Cache Size	1% of the catalog (BAYHAN et al., 2016)
Link Latency	30ms (GARCIA-LUNA-ACEVES; MIRZAZAD-BARIJOUGH; HEMMATI, 2016)
Link Bandwidth	10Gbps (GARCIA-LUNA-ACEVES; MIRZAZAD-BARIJOUGH; HEMMATI, 2016)
Mobility Pattern	Session-Disconnection Cycles (TUDUCE; GROSS, 2005)
Mobility Cycle Period	5 minutes (300 seconds)
Producer Availability	30% to 100%
Production Period	90 seconds
Catalog Size	1000 objects (BAYHAN et al., 2016)
Object Popularity	Zipf distribution $\alpha = 0.8$ (BAYHAN et al., 2016)
Object Size	1000 chunks (BAYHAN et al., 2016)
Total Interest Requests	968.000 (BAYHAN et al., 2016; MICK; TOURANI; MISRA, 2016)
Warm-up Requests	240.000 (BAYHAN et al., 2016; MICK; TOURANI; MISRA, 2016)
Evaluation Requests	728.000 (BAYHAN et al., 2016; MICK; TOURANI; MISRA, 2016)
Request Rate	Poisson process $\lambda = 0.02$
Retransmission	225 seconds (MASTORAKIS et al., 2015).
Evaluation Period	75 minutes
Vicinity Size	2 (LEHMANN; BARCELLOS; MAUTHE, 2016)
Replication Degree	1 (LEHMANN; BARCELLOS; MAUTHE, 2016)
Placement Policy	Random (LEHMANN; BARCELLOS; MAUTHE, 2016)
Mobile Providers	20
Providers Availability	80% to 100%
Custodian Storage Size	20% of the catalog
Unsolicited Data Pushed	10 objects

its content to consumers. Its availability ranges from 30% to 100% in each execution and determines the ratio between session and disconnection times in mobility periods that last 5 minutes. For instance, a producer with 60% availability has succeeding cycles of session and disconnection (TUDUCE; GROSS, 2005) that last, on average, 180 and 120 seconds respectively. Lastly, the producer also creates a new object periodically every 90 seconds that replaces an existing one from the catalog, keeping its size constant (ELAYOUBI; ROBERTS, 2015).

The catalog has a fixed size of 1000 1MB objects with popularity following a Zipf distribution (α = 0.8) (BAYHAN et al., 2016). During the simulation, 968k Interest requests are issued, divided into 240k in the warm-up period and 728k in the evaluation period (BAYHAN et al., 2016; MICK; TOURANI; MISRA, 2016). The requests for objects are sent following a Poisson process (λ = 0.02), resulting in 1 new object request every 50 seconds. In the case of a failed request, consumers detect a timeout and retrans-

Figure 5.1: Network topology used in the evaluation



mit it again after 225 seconds[3], approximately the maximum retransmission timeout of ndnSIM (MASTORAKIS et al., 2015). In summary, 728 objects composed of 1000 1KB chunks each are requested by 8 different consumer streams during the evaluation period, which lasts 75 minutes.

In our evaluation, the ratio between content creation and request is 1.8, which means that few requests happen between the generation of new objects, resulting in a high replication cost. If the ratio increases, the replication cost would be proportionally less relevant and benefit a larger number of requests. Moreover, NDN through cache and request aggregation would naturally operate better in disseminating content given the popularity distribution and also benefit PDRM. Therefore, we use this ratio in the analysis without loss of generality.

Each of the three mechanisms that extend the Vanilla NDN has internal parameters that affect its performance. For object replication, PDRM uses a vicinity size of 2, replication degree of 1, and random placement policy (LEHMANN; BARCELLOS; MAUTHE, 2016), and has 20 mobile users with availability ranging from 80% to 100% connected

---

[3] Notice that ndnSIM updates the retransmission timeout per chunk, doubling the estimate RTT after a timeout from the minimum value of 200ms to the maximum value of 200 seconds. Since our consumer application requests all 1000 chunks together, it will probably either retrieve or fail them all. In the case of failure, instead of having different retransmission timeouts for the first chunks, we decided to set a single value for all chunks of an object.

to the access router. They can become content providers on behalf of the producer but do not send periodic requests like the consumers. Data Depot uses two configurations of a custodian connected to the access router. In the first, the custodian stores the last 1000 produced objects (100% of the catalog), which is similar to a CDN. In the second, we limit the custodian storage to approximate to the PDRM performance in that scenario. Lastly, a producer using Data Spot can send up to 10 objects to the access router (its cache size) as unsolicited data.

5.3 Metrics

We analyze both the performance and overhead of each proposal to support content mobility during the evaluation period over several runs. The results presented in the following section for each metric collected are based on the average of all runs. The following metrics have been measured and evaluated.

Object Download Time. The average time elapsed since consumers request the first chunk and retrieve the 1000th (last) chunk of an object. This metric is affected by the content unavailability, which causes timeouts and re-issue of pending chunks. The download time represents content availability in the network perceived by the consumers. The goal of each proposal is to benefit consumers by reducing their object download time.

Served Data Ratio. The average percentage of chunks served by each existing entity in the network. The result of a chunk request can be either failure (i.e., never retrieved) or success. In the case of success, one of the many potential providers may have served the chunk: producer, routers, or a mechanism-specific element (e.g., providers and custodian). The goal of each proposal is to reduce the producer load, relying more on routers and other providers (when available) to serve the requested content.

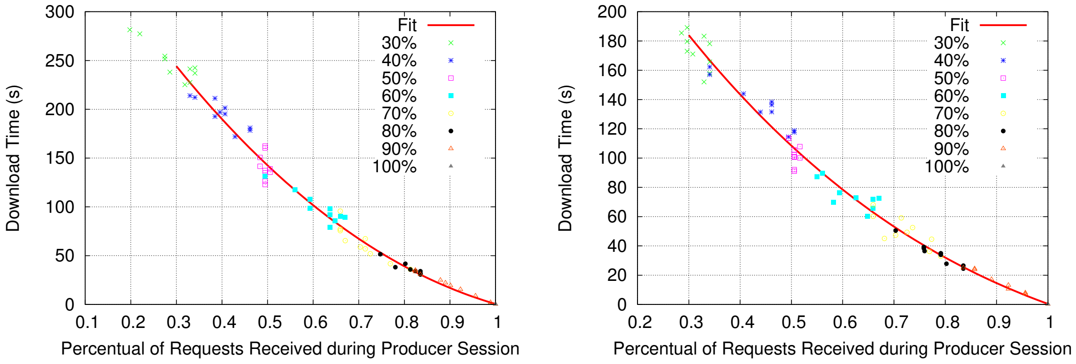
Extra Traffic Generated. The amount of traffic generated by the MP beyond satisfying Interest requests from consumers. That is, it measures the overhead traffic from control and signaling messages (Interest packets) as well as pushed objects (Data packets). The goal of each proposal is to reduce the traffic generated, particularly the Data packets which consume more resources.

FIB changes. The number of modifications in the routing tables required by each proposal. Some of the proposals rely on announcements on behalf of the producer and dynamic topological adjustments, which causes an overhead to the network concerning routes recalculation and FIB reconfiguration. The goal of each proposal is to cause the least amount of topological changes to the network to avoid too many FIB recalculations, which can be costly.

5.4 Producer Availability

The average producer availability is a simulation input that describes how much time the producer stays connected to the network providing content. However, the consumers through their requests may perceive a different producer availability than the defined by the input. For example, consider a producer with 90% average availability over multiple simulation runs. On one extreme, the producer can be connected when every consumer request is issued. In this case, the consumers see a producer always available, which achieves optimal download performance and better than the expectation. On the other extreme, the same producer could receive only 80% of the requests while connected, leading to a worse download performance than expected.

Figure 5.2: Scatter plot and fit curve of the mismatch between the producer availability expected by the input and the one perceived by the consumers.



(a) Vanilla NDN

(b) PDRM

Figure 5.2 evidences the mismatch between the producer availability expected by the input and the one perceived by the consumers. Numerically, we observed that the perceived average might deviate roughly from minus 10% to plus 10% around the expected average. The two scatter plots in Figure 5.2 contain the results of all single runs in the uniformly distributed content scenario using Vanilla NDN and PDRM. They represent the average download time of consumers (y-axis) given the producer availability perceived by consumers (x-axis). Each point of the plot also specifies, according to shape and color, the input producer availability, which varies from 30% to 100%. Lastly, the curve presented is a fitting of the points to a third-degree function.

A thorough analysis of all simulation runs showed that this behavior is consistent across all producer availability values, scenarios, and mechanisms evaluated. They also indicate that the producer availability observed by the consumers is more accurate for the analysis than the input value for performance evaluation. We fit a third-degree function in the points of the scatter plots to represent the mechanism performance behavior according to the producer availability perceived by the consumers. The analysis of the results presented in the following section will be according to the fitting curves generated based on the measured data.

6 RESULTS

This chapter presents and discusses the results obtained from the PDRM evaluation. It contains a total of four scenarios divided into two parts, with two scenarios each. The first part focuses exclusively on the PDRM operation and measures the impact of the number of available providers and in-network cache capacity on its performance. The second part compares PDRM to Vanilla NDN, Data Depot, and Data Spot in two scenarios with periods of producer unavailability. The difference between these two scenarios is how content is distributed: uniformly or clustered in areas of interest.

6.1 Number of Available Providers

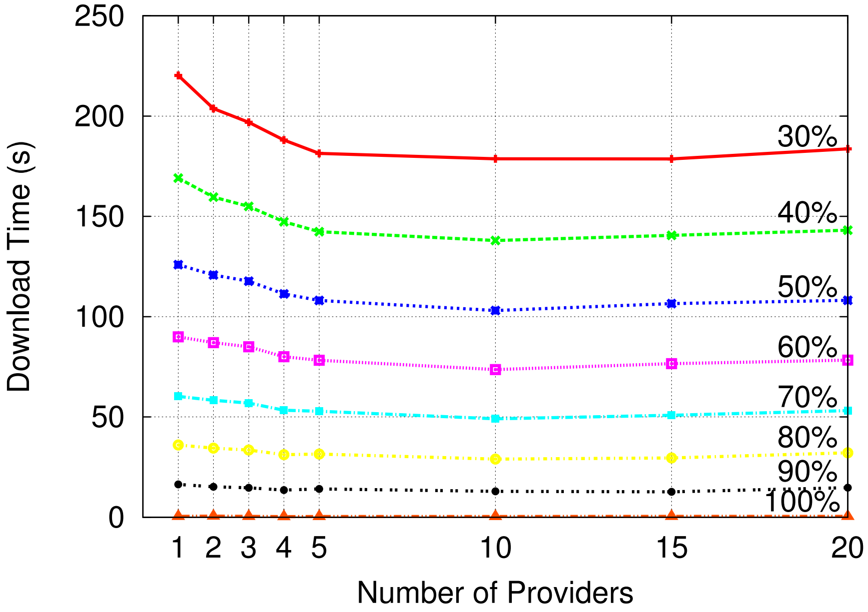
The first scenario evaluates the impact of the number of potential providers on PDRM because the mechanism relies on them to replicate content. Intuitively, PDRM replication and performance will increase directly with the number of providers available. In this analysis, we measure the effects of the provider pool size on the consumer download time, replication percentage, and the fraction of total requests satisfied by the network elements. The results consider a varying number of potential providers in the vicinity between 1 and 20.

We start presenting the average download time of consumers given the number of available providers in the vicinity, shown in Figure 6.1. There are 8 curves, each representing the producer availability between 30% and 100% with a step of 10%. The results show the impact of both the producer availability (when comparing the curves) and the number of potential providers (when analyzing each curve individually) on the consumer download time.

The results evidence that the producer availability has more impact than the number of providers on the consumer download time. The difference between the set of content objects served by the producer and providers justifies this result. The producer has the entire content catalog and can serve any object requested while the providers possess only a small fraction of the content catalog (i.e., the most popular objects). Therefore, the producer remains essential for serving the non-replicated objects.

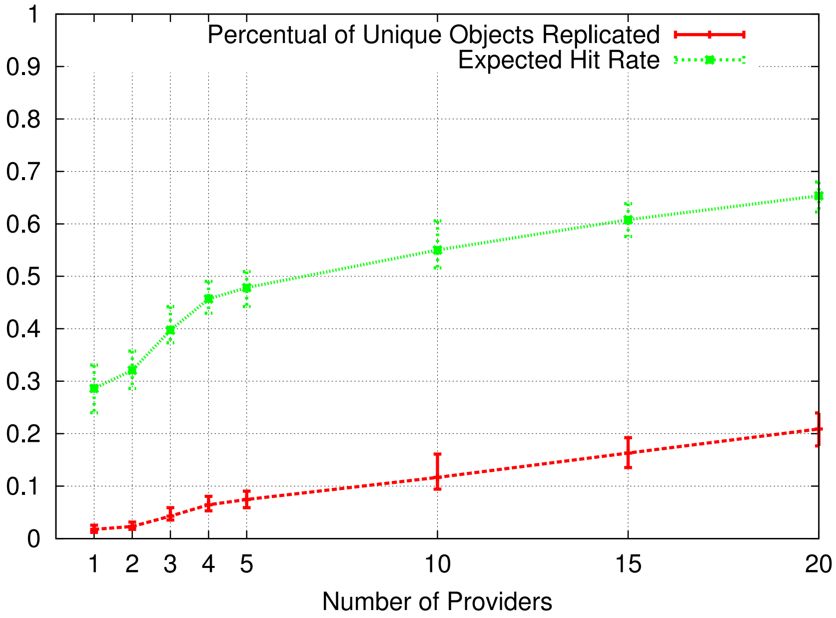
Although not the most impactful parameter, the number of potential providers still affects the download times. As the number of providers increases so does the probability of object replication because PDRM only replicates content objects that are interesting for

Figure 6.1: Consumer download time according to the number of potential mobile providers using PDRM



at least one potential provider. As expected based on this design, the longest download times occur when there is a single potential provider in the vicinity besides the producer. As the number of potential providers increases, the download time gradually decreases until it converges at 10 providers, maintaining a similar performance afterward. In all cases, the PDRM performance improves between 16.5% and 24% by having more potential providers.

Figure 6.2: Percentual of unique objects replicated and expected hit rate according to the number of potential mobile providers using PDRM



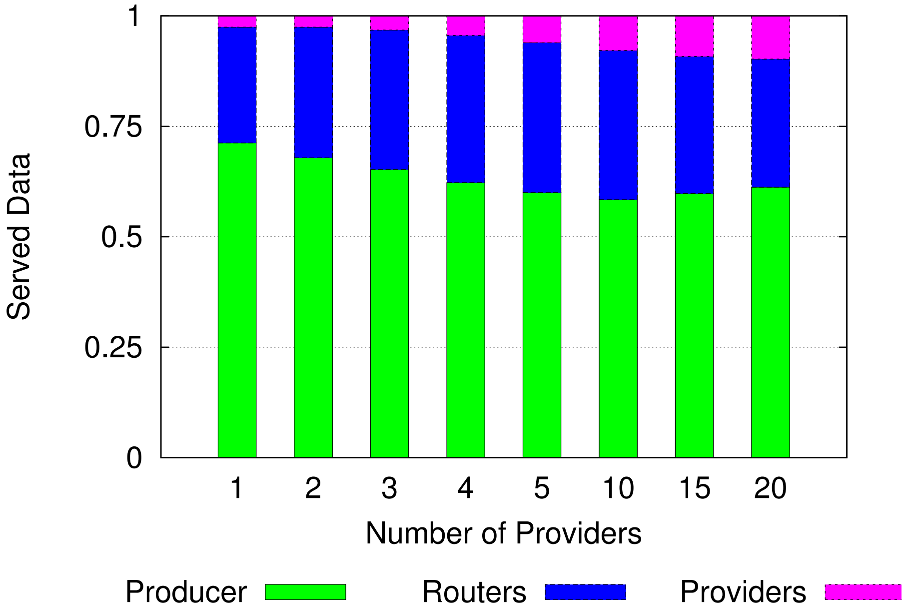
Next, we investigate further the relation between the number of providers and replication degree, and how they affect the download time. Figure 6.2 presents the percentage of unique objects replicated and the expected hit rate according to the number of potential providers. We observe that the replication percentage increases while the expected hit rate decreases logarithmically as the number of potential providers increases, a consequence of the Zipf-like popularity distribution that has few objects very popular and a long tail of unpopular ones. Since the replication cost for each object is constant, PDRM faces a trade-off between the overall cost of replication and the benefits obtained from the set of content objects replicated.

With a single provider in the vicinity, PDRM replicates only 1.76% of the catalog caused by the provider interest in the catalog, expecting the provider to satisfy 28.6% of the requests given the popularity of the top 1.76% objects. As the number of potential providers increases, these metrics also do, but at different rates, evidencing the trade-off. With 5 providers, PDRM achieves a 7.5% replication percentage and a 47.8% expected hit rate. Moreover, with 10 providers in the vicinity, PDRM replicates 11.6% of the catalog, which covers 55% of the expected requests. Lastly, having 20 providers, we observe a high replication increase but the expected return is not as significant as before. The replication percentage almost doubles, from 11.6% to 20.8%, but the expected hit rate increases merely 10%, from 55% to 65.3%.

The download times results combined with the trade-off analysis between replication cost and expected hit rate indicates a saturation point in which PDRM achieves the lowest download times and replicating more objects does not improve performance. Having fewer providers diminishes the chance of users willing to provide content, resulting in low replication. In contrast, with more providers, the probability of finding someone to receive replicated objects (even unpopular ones) is higher. However, the download times and expected hit rate growth show that replicating a larger set of objects does not improve the performance significantly. Besides, it also increases the producer overhead and generates traffic of unpopular content, which can pollute the caches and affect their operation negatively.

Finally, we analyze the impact of the numbers of providers in the contribution rate of each network entity, shown in Figure 6.3. We present the average values of all producer availability levels because there is a maximum 5% variation in the percentage of content served. The results show that when using PDRM with only 1 provider in the vicinity, the producer serves on average 71.25% of the requests while the remainder is satisfied by the routers (26.21%) and provider (2.54%). This result indicates two benefits of replicating popular content to reduce the producer load: it directly creates new copies on end users and indirectly creates new copies on routers.

Figure 6.3: Providing source distribution according to the number of potential mobile providers using PDRM



When more potential providers are present, their participation in data distribution increases. In numbers, they serve 6.07%, 7.83%, and 9.76% of the requests respectively for 5, 10, and 20 providers because of the low replication percentage of objects in this scenario, at most 20.8% of the catalog. The producer and router loads follow the trend discussed during the download time analysis. The producer load decreases to 60% (5 providers) and 58.41% (10 providers) while the routers load increases to 33.93% (5 providers) and 33.76% (10 providers) as the number of providers increase until its saturation point. From this point on, the percentage of served content by the routers decreases because of the potential pollution of caches with unpopular objects when replicating a larger amount of content. The replacement of popular objects in the caches reduces their effectiveness and, consequently, increases the producer load, as observed with 20 providers: the producer load increases to 61.23% whereas the router load decreases to 29.01%.

Through the analysis of this scenario, we observe that PDRM requires a certain pool of providers to operate properly. Ideally, the number of potential providers and replication percentage should not be too low or high to achieve the best results. The best values vary according to the scenario configuration and are affected mostly by the content popularity distribution, consumer request rate, and cache capacity. For the remainder of the evaluation, we use 20 mobile providers to investigate the impact of proactive replication.

6.2 In-Network Cache Size

The second scenario is used to measure the impact of available in-network caching resources on the content mobility support provided by PDRM. The goal of this analysis is to establish the importance of mobile providers compared to the in-network caching, a major factor for content dissemination in NDN. Similar evaluations in the past have considered cache capacities varying between 0% and 10% of the catalog (BAYHAN et al., 2016; MICK; TOURANI; MISRA, 2016). We analyze PDRM with cache size values of 0%, 1%, 2%, and 5% (MICK; TOURANI; MISRA, 2016).

Figure 6.4: Consumer download time according to the in-network cache sizes using PDRM

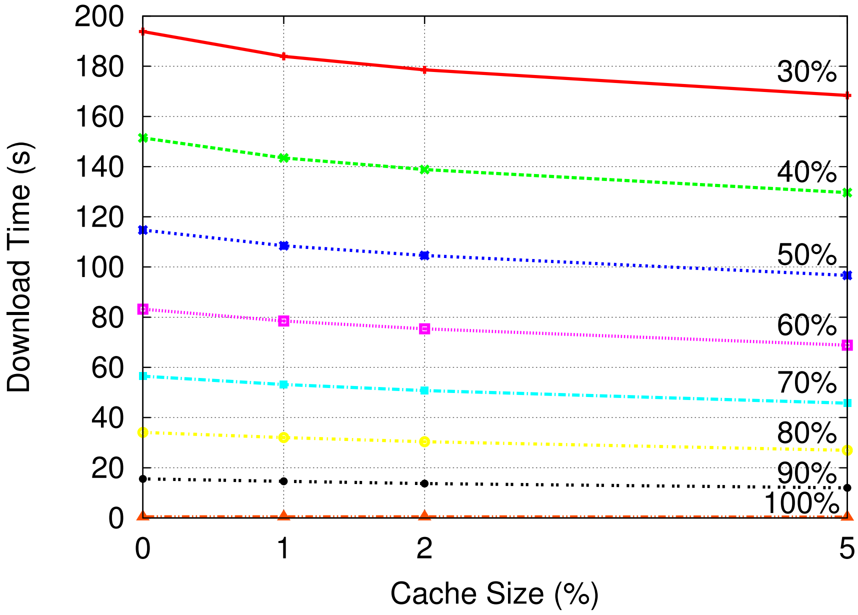


Figure 6.4 presents the download time of consumers when the producer uses PDRM according to four cache sizes. Similar to the previous scenario, there are 8 curves, each representing the producer availability between 30% and 100% with a step of 10%. The results allow us to analyze the impact of both the producer availability (curve comparison) and the cache size (analysis of curves individually).

Similar to the previous scenario, Figure 6.4 shows that producer availability impacts more than cache size on download times. We also notice, unsurprisingly, that more in-network cache will reduce average download times. The performance improvement achieved by PDRM can be examined in absolute or proportional values. They present opposing behaviors: the absolute improvement is higher with lower producer availability while the proportional one achieves better results with higher producer availability (except when always available).

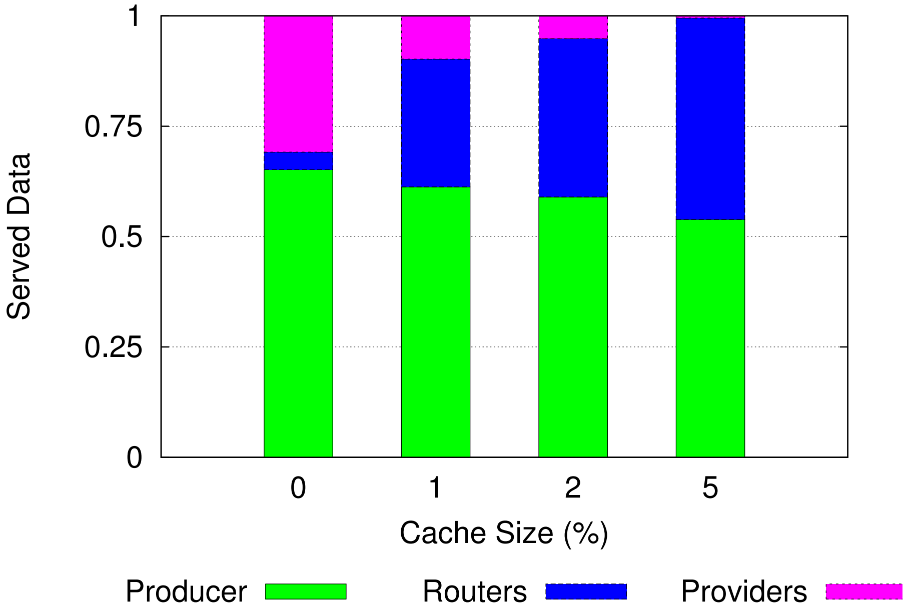
This behavior occurs due to the room for improvement in each case. Without cache, consumers rely solely on the producer to obtain content. A producer with low availability leads to poor download performance while a producer highly available tends to significantly better download times. The addition of cache to the network boosts the content mobility support for producers according to their room for improvement. For instance, between 0% and 5% cache sizes, PDRM reduces the download times from 193.80 to 168.34 seconds (30% availability), from 83.23 to 68.84 seconds (60% availability), and from 15.56 to 11.99 seconds (90% availability). Despite a diminishing absolute reduction (25.46, 14.39, and 3.57 seconds), the proportional download improvement increases in these cases: 13.14%, 17.29%, and 22.93%.

We use Figure 6.5 to understand further the impact of in-network cache size on PDRM. It shows the fraction of requests served by each element (producer, provider or routers) according to cache size and producer availability. The results indicate that the providers and routers complement themselves in reducing the producer load of content dissemination. That is, when the routers are less effective (smaller cache sizes), the providers have a bigger role in serving content. When the routers have a higher impact (larger cache sizes), the providers have only a marginal direct contribution disseminating content. In this case, they also have an indirect positive impact by populating the caches with popular content as a side-effect of replication.

Across all cache configurations, PDRM keeps the producer load in average between 65.18% (0% cache) and 53.85% (5% cache). With 0% cache, consumers rely mainly on the producer or providers to serve content because of the lack of cached objects by the routers. They can still help the dissemination through request aggregation, a feature in which multiple data responses are distributed with a single interest request, but it has limited effectiveness. The numbers confirm this intuition, showing that providers serve 30.84% of the requests while routers contribute with only 3.96% with 0% cache.

When caches are added to the network, routers can be more effective in disseminating content. Consequently, the providers reduce their direct participation in the content dissemination but increase their indirect impact by populating in-network caches. Numerically, the percentage of content served by them decreases gradually as the cache

Figure 6.5: Providing source distribution according to the in-network cache size using PDRM



size increases: 9.77%, 5.17%, and 0.49% respectively for 1%, 2%, and 5% cache sizes. Meanwhile, the routers extend their share in the content dissemination, serving 28.96%, 35.83%, 45.64% of the requests respectively for 1%, 2%, and 5% cache sizes.

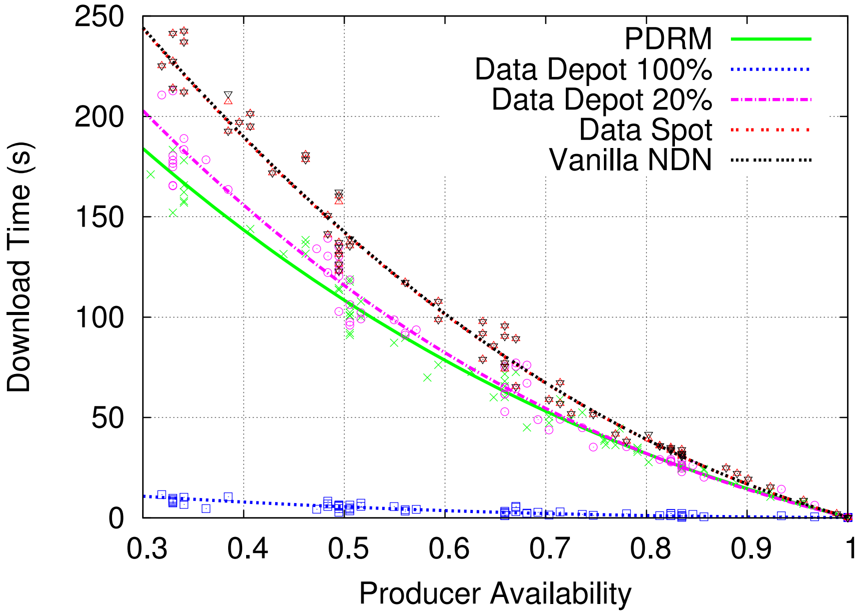
This analysis shows that PDRM has two effects regarding NDN native in-network caching: serves data on behalf of routers if they are unable and boosts their cache performance when available. The impact of PDRM is inversely related to the amount of cache available in the network, being more significant the lower the cache sizes are. Studies in the ICN literature usually consider a small cache (around 1%) (BAYHAN et al., 2016; MICK; TOURANI; MISRA, 2016), as the most representative cases. This configuration results in high competition from objects for the limited cache resources, which benefits PDRM performance greatly. We assume a cache capacity of 1% of the catalog for the remaining two scenarios.

6.3 Uniformly Distributed Content

This scenario begins the comparison of PDRM to Vanilla NDN and two state-of-the-art proposals. The mechanisms are evaluated for different producer unavailability periods, from 30% to 100%, under the simplifying assumption that content interest is uniformly distributed. Both caching and replication are negatively affected by this assumption, so we expect the results of all proposals to suffer in varying degrees, representing roughly a lower boundary of their performance. In this evaluation, the custodian of the Data Depot has two capacities: 200 and 1000 objects (20% and 100% of the catalog respectively).

Figure 6.6 shows that PDRM achieves an average download time of 51.983 seconds, outperforming Vanilla NDN, Data Spot, and Data Depot with 200 storage capacity. PDRM reduces the download times by 23.13% compared to the first two and by 6.04% compared to the last. As expected, results show that in the ideal case that Data Spot can store the entire catalog of objects remotely then it achieves the best results, taking only 2.627 seconds on average to retrieve content.

Figure 6.6: Consumer download time in the uniformly distributed content scenario using PDRM, Data Depot, Data Spot, and Vanilla NDN

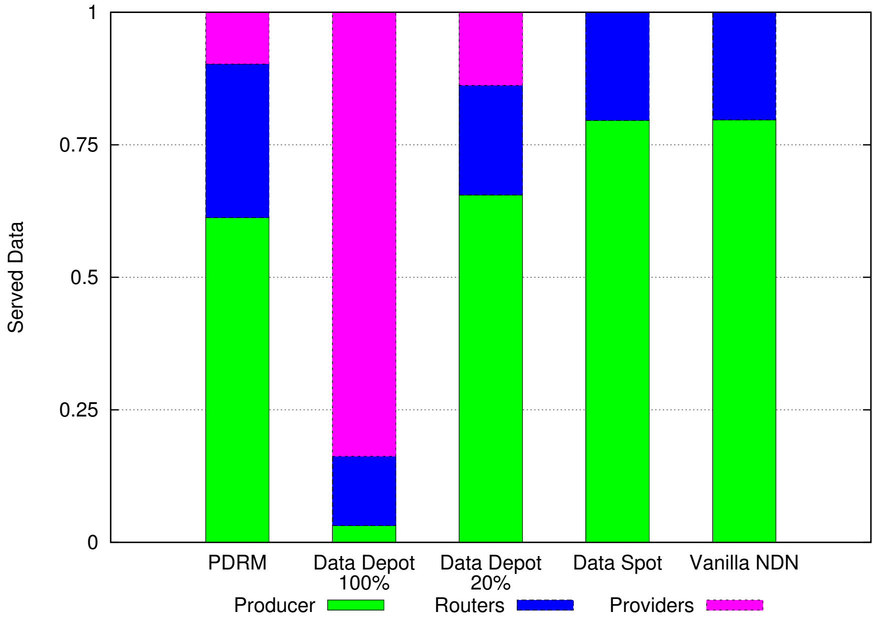


The better performance of PDRM and Data Depot are a consequence of the proactive content replication of produced objects. The replication decisions used by each mechanism explain the gap between them. PDRM probes the vicinity to infer which content objects are popular in which regions. Since this scenario has a uniformly distributed content popularity, PDRM fails to benefit from it entirely, replicating only 20% of the catalog. The Data Depot mechanism, in its turn, replicates every content and replaces them in the storage according to their lifetime. Due to the large storage capacity, the custodian can serve many objects using this brute force approach.

Data Spot and Vanilla NDN have the worst download performance because they rely mostly only on the reactive in-network caching. The poor results achieved by these mechanisms indicate that proactive replicating before the producer unavailability or consumer requests is beneficial for the download performance in a mobile environment. Regarding the Data Spot, the size of objects used in the simulation harms its operation. In this analysis, content objects are small and retrieved entirely in a single RTT. Hence, the Data Spot mechanism does not have many opportunities to push pending requests to the routers and has a similar outcome of Vanilla NDN.

Figure 6.7 shows how each mechanism impacts on the load of the elements in the network, shedding light on the reasons they achieve the download performance presented previously. The producer using PDRM has the second lowest producer load among all proposals, serving 61.26% of the total requests. As discussed in the previous evaluations, the direct (providers) and indirect (routers) effects of PDRM combine for this. In numbers, routers serve 28.96% of the requests while the providers satisfy the remainder 9.78%.

Figure 6.7: Providing source distribution in the uniformly distributed content scenario using PDRM, Data Depot, Data Spot, and Vanilla NDN



Data Depot obtains the first and third best results regarding producer load with its two configurations. Because the custodian is always available and can store the full content catalog, it serves 83.74% of requests on behalf of the producer. The producer (3.16%) and the routers (13.10%) split the remainder of the requests. The Data Depot with 200 storage capacity has a slightly worse result compared to PDRM. It reduces the producer load to 65.53%, relying more on the custodian than PDRM on their providers. In this configuration with limited storage, Data Depot does not perform greatly because the producer pushes every produced object to the custodian, independently of their popularity.

Therefore, it fails to populate the caches with popular content like PDRM. Consequently, the custodian serves 13.78% of requests while the routers 20.69%, more than 8% less compared to PDRM.

Vanilla NDN and Data Spot rely only on the producer and routers to distribute content, even when objects are pushed unsolicitedly. Different from the other mechanism, Data Spot does not use end users to help the dissemination. The results show that they have nearly the same load distribution with an insignificant variation of 0.08%. The producer-router ratio of served data is around 79.65%-20.35%. Assuming the Vanilla NDN results as a baseline, we observe that both PDRM and Data Depot reduce the producer load significantly and that PDRM is the only mechanism to improve the cache efficiency in the network.

Table 6.1: Overhead summary of the mechanisms in the uniformly distributed content scenario using PDRM, Data Depot, and Data Spot

Proposal	Data	Interest	FIB Changes	Improvement
PDRM	11,312.97	10,713.96	10.67	23.13%
Data Depot 100%	52,464.91	52,412.50	104.82	96.12%
Data Depot 20%	52,464.91	52,412.50	104.82	18.19%
Data Spot	462.50	0.00	0.00	0.26%

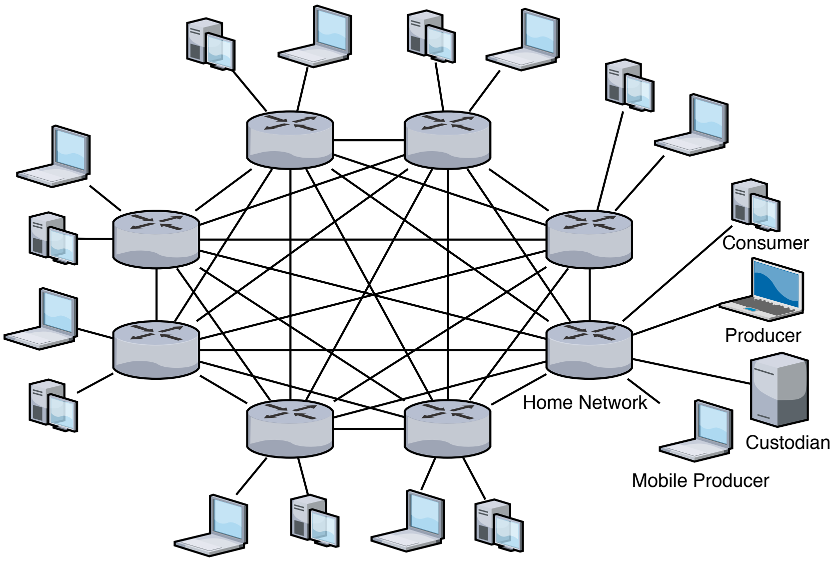
Lastly, the analysis discusses the overhead produced by each mechanism to obtain their results. Table 6.1 presents the number of Data and Interest packets generated, FIB changes, and the download improvement compared to Vanilla NDN. As expected, the idealized scenario of Data Depot with full catalog provides the best results by far. Otherwise, PDRM achieves the best results, while generating only 20% of the packets and requiring 10% of the topological adjustments made by Data Depot. The difference is that PDRM selects which content objects are worth replication instead of replicating every produced object. This decision reduces overhead to the network and improves the download performance by focusing the resources mainly on popular objects. The custodian storage capacity does not impact the overhead of Data Depot because the producer pushes the content regardless. Nevertheless, it impacts significantly on the download performance because of the amount of replaced objects when the storage is full. The overhead of the Data Spot mechanism corroborates its low efficacy demonstrated in this analysis, replicating less than 1 object in average.

6.4 Content Clustering

The last scenario studies the performance of the mechanisms to support content mobility under producer unavailability periods and content locality. More precisely, consumer requests are determined by their locality in a network composed of multiple domains. PDRM is designed to take advantage of the location by probing devices in the vicinity and pushing content where it is popular. Consequently, PDRM is expected to outperform the other proposals by placing content objects closer to consumers.

The network topology is a complete graph with 8 nodes, each representing a distinct domain, as illustrated in Figure 6.8. The producer has a home network and moves randomly between the domains. When executing PDRM, there is one potential provider in each domain, totaling 8 in the network. In the case of the Data Depot mechanism, the custodian is located in the home network of the producer and has a capacity of 1000 and 700 objects (100% and 70% of the catalog respectively).

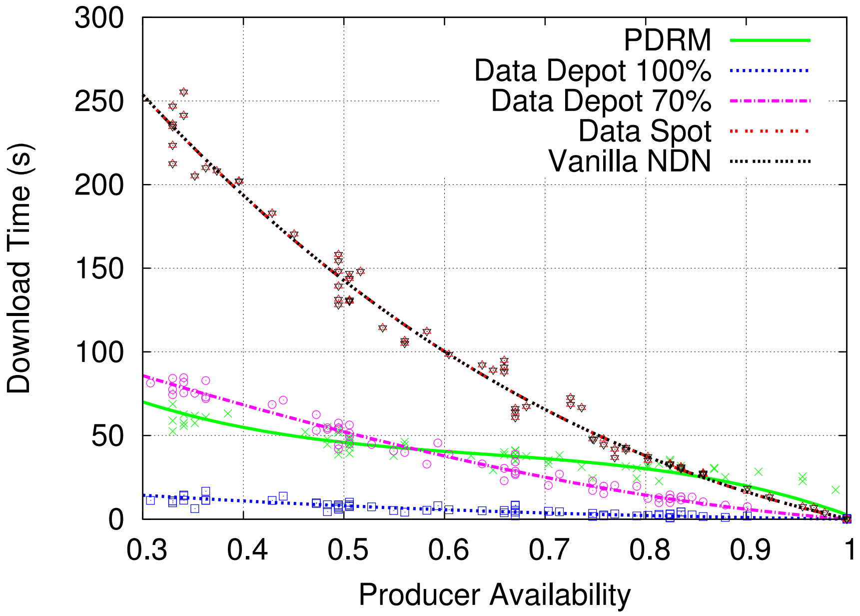
Figure 6.8: Network topology used in the content clustering scenario



The catalog has 1008 objects, divided into 9 sets containing 112 objects each. They represent the local content of each domain and the global content independent of domains. In each domain, there is a single consumer stream that requests both local and global content in a proportion of 75% and 25% (BAYHAN et al., 2016). In addition to the other metrics, we measure the inter-domain traffic, defined by the amount of traffic flowing between domains.

Figure 6.9 shows the download times achieved when the producer uses each mechanism. On average, PDRM outperforms Vanilla NDN and Data Spot by 61.09%. PDRM

Figure 6.9: Consumer download time in the content clustering scenario using PDRM, Data Depot, Data Spot, and Vanilla NDN



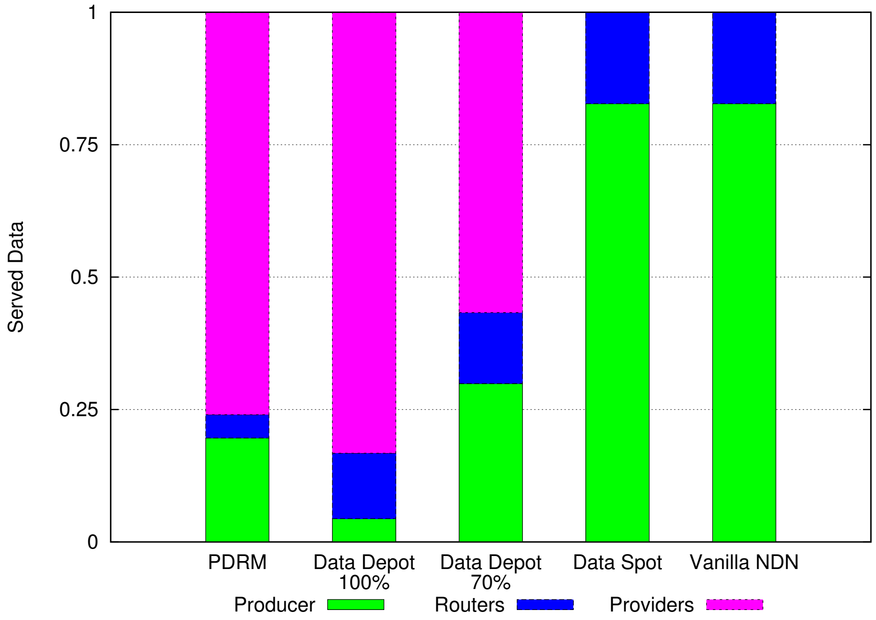
boosts the download performance especially with lower producer availability because the native caching fails to provide adequate content mobility support. Overall, PDRM takes advantage of content locality in two aspects: it learns about the interested consumers and pushes objects towards them, and it replicates a larger fraction of objects due to the catalog division.

PDRM also achieves similar download performance as Data Depot with a custodian storing up to 700 objects. The average download times across all configurations are 26.424 (PDRM) and 24.572 (Data Depot 70%) seconds. PDRM performs better with lower producer availabilities (less than 55%) but worse with higher ones. This behavior is a consequence of the replication strategies adopted by the mechanisms. PDRM replicates popular objects while Data Depot replicates every content object. This brute force approach may pollute the custodian and caches with unpopular content, reducing their performance. As expected, PDRM performs significantly worse than Data Depot using a custodian that stores the entire catalog. This case represents a CDN, in which content is always available on the network, resulting in an average download time of just 3.846 seconds compared to 26.424 seconds obtained using PDRM.

Figure 6.10 shows the providing source distribution achieved by each mechanism and helps to explain the download time results. When the producer uses PDRM, its load is only 19.64%. Different from the previous scenario, the PDRM proactive replication shows its full effectiveness: the majority of requests are served by the providers (75.96%)

rather than by the routers (4.40%). This outcome corroborates that PDRM helps the content dissemination by identifying popular content as well as pushing copies to potential providers and towards interested consumers.

Figure 6.10: Providing source distribution in the content clustering scenario using PDRM, Data Depot, Data Spot, and Vanilla NDN



Data Depot achieves a producer load even lower than PDRM: 29.90% (70% storage size) and 4.40% (100% storage size). The custodian is always available to serve content and has a storage large enough to store a significant set of objects. As a consequence, the custodian satisfies 56.69% and 83.25% of the requests, leaving the remainder 13.41% and 12.35% for the routers, respectively for 70% and 100% storage sizes. The results show that the producer using Data Spot and Vanilla NDN has a load of 82.74%, leaving routers to serve 17.26% of the data. The network topology explains the bad cache efficiency in disseminating content. Each domain has a single level, responsible for both caching and inter-domain routing. Since routers have to forward requests and responses from other domains, they pollute their caches with data not interesting or requested locally.

The overhead of the mechanisms is presented in Table 6.2. PDRM is the second most consuming mechanism, generating almost 40,000 additional Data and Interest packets while requiring around 39 FIB changes. Compared to Data Depot, the mechanism with the highest overhead, PDRM reduces the number of packets generated and FIB changes in 25% and 60% respectively, independently of the storage size of the custodian. Data Spot pushes fewer copies because of the small-sized objects, reducing the chance of pending requests when starting a movement. Consequently, it generates little overhead.

Table 6.2: Overhead and data traffic summary of the mechanisms in the content clustering scenario using PDRM, Data Depot, Data Spot, and Vanilla NDN

Proposal	Data	Interest	FIB Changes	External Traffic (% of all traffic)
PDRM	39.4k	39.8k	39.02	336.47 MB (41.83%)
Data Depot (100%)	52.4k	52k	104.42	593.50 MB (77.75%)
Data Depot (70%)	52.4k	52k	104.42	601.85 MB (78.84%)
Data Spot	400.00	0.00	0.00	581.00 MB (76.11%)
Vanilla NDN	0.00	0.00	0.00	581.05 MB (76.12%)

Finally, we examine the external traffic that each proposal generates, also shown in Table 6.2. The primary goal is to keep traffic within the domain to reduce cost and improve consumer performance. PDRM has the lowest inter-domain traffic among all proposals, with 336MB, representing 41.83% of the total traffic in the network. The explanation for this result is the combination of higher replication percentage and pushing copies towards the consumers. The other three proposals generate around 581MB, which is 76.12% of the total traffic. Noteworthy, Data Depot results in more inter-domain traffic, between 601.85 and 593.50MB, because the producer always sends data to the custodian when replicating.

6.5 Summary

In the PDRM evaluation, we investigated the PDRM operation and performance under several aspects in four scenarios. First, we quantified the effects of various mobile providers and in-network cache sizes on the producer, consumers, and network. Then, we measured the PDRM performance compared two Vanilla NDN and two state-of-the-art proposals, namely Data Depot and Data Spot, in two scenarios with periods of producer unavailability: one with content uniformly distributed and the other with content clustered in areas of interest.

The first analysis indicates that on one extreme, replicating too few objects may not achieve the full benefits of PDRM. On the other extreme, replicating too many may pollute the caches with unpopular content. PDRM improves the download performance gradually when adding the first extra content providers until converging the results when the number of providers saturates. At this point, which varies according to the network configuration, PDRM may even fail to achieve its full benefits because of over replication that potentially pollutes the caches with unpopular content.

The second analysis indicates that PDRM is a complementing feature of NDN's in-network caching. That is, the results show that PDRM works proportionally inverse to in-network caching. Consequently, PDRM together with NDN provides consistent download performance independent on the available cache in the network.

The third scenario can be seen as a lower boundary for performance because of the simplifying assumptions regarding content location. PDRM reduces the download times compared to Vanilla NDN and Data Spot by 23.13% and Data Depot with 200 storage capacity by 6.04%. Further, PDRM also reduces the producer load to 65.53%, the second best among all proposals. As expected, Data Depot with 1000 storage custodian performs significantly better than any other proposal because it represents a CDN in which content is always available. Nevertheless, PDRM generates 20% of the Data Depot overhead, sending less than 12,000 Data and Interest packets and requiring around 11 FIB changes.

The fourth scenario can be seen as an upper boundary for PDRM performance because it enables the mechanism to evidence its full potential to support content mobility. Compared to Vanilla NDN and Data Spot, PDRM reduces the download times up to 61.09%, producer load up to 76.26%, and inter-domain traffic up to 46.50%. It also achieves similar performance to Data Depot when the custodian has 700 storage capacity but performs worse than Data Depot acting as a CDN. However, PDRM requires no prior storage investment, consumes fewer resources, and may perform better under certain conditions.

7 CONCLUSION

In this thesis, we investigated how to use proactive and location-aware data replication to improve the content mobility support in NDN. As the result of the investigation, we proposed the Proactive Data Replication Mechanism (PDRM) to increase content availability through data redundancy maintaining efficient use of network and end-user resources. PDRM differs from other state-of-the-art proposals by learning about its vicinity and using this information to influence the replication decisions. The learning step is the key feature that allows PDRM to optimize the content replication process for high dissemination performance and low resource consumption.

PDRM achieves these objectives through three key parameters: vicinity size, replication degree, and placement policy. PDRM uses the smallest vicinity that contains potential mobile providers. It enables PDRM to replicate content while also limiting the overhead generated close to the producer. PDRM replicates only those objects in which users are interested in consuming and becoming potential providers with only one replica. These decisions reduce the waste of resources on a costly pushing operation and yield the best cost-benefit ratio. Lastly, PDRM selects a random user from the pool of potential providers with two purposes in mind. First, it distributes the load on mobile providers, avoiding overloading any that could be considered the best provider. Second, obtaining reliable information regarding other users in a mobile environment is very costly and proved not worth it.

Throughout the thesis, we evaluated PDRM over different aspects. First, we evaluated the impact of the number of available providers and in-network cache capacity on PDRM performance. The conclusions found are that PDRM improves the download performance gradually when adding the first extra content providers until converging the results when the number of providers saturates. At this point, which varies according to the network configuration, PDRM may even fail to achieve its full benefits because of over replication that potentially pollutes the caches with unpopular content. Regarding the in-network cache capacity, we show that PDRM works proportionally inverse to it, providing constant performance independent on the available cache.

Then, we concluded the PDRM evaluation with a comparison between PDRM, Vanilla NDN, and two state-of-the-art proposals, namely Data Depot and Data Spot, in two scenarios with periods of producer unavailability. The first one has content uniformly distributed in the network and represents a lower bound for performance. PDRM reduces the download times by between 23.13% (Vanilla NDN and Data Spot) and 6.04% (Data Depot), decreases the producer load to 65.53% while generating an overhead 80% lower than Data Depot. The second scenario uses clustered content around areas of interest, representing an upper performance bound. PDRM evidences its full potential in this scenario. Compared to Vanilla NDN and Data Spot, PDRM reduces the download times up to 61.09%, producer load up to 76.26%, and inter-domain traffic up to 46.50%. As expected, the CDN case (Data Depot with a custodian storing the full catalog) outperforms the remainder mechanisms in both scenarios. In the first scenario, PDRM has similar results to a custodian storing 200 objects while in the second 700 objects but using significantly fewer resources and generating smaller overhead.

Overall, the evaluation indicates promising results for proactive replication in the context of mobility, despite this approach not being suggested in regular networks without mobility (SHARMA; VENKATARAMANI; SITARAMAN, 2013; FAYAZBAKHSH et al., 2013). As for future work, there are three extension directions: refinement, evaluation, and deployment. PDRM can be refined to infer better content popularity and how to address it to avoid over replication as well as its economics and how to operate with multiple producers competing for the provider resources. The second direction considers how to evaluate PDRM further: different scenarios, such as real-time or streaming applications, and mobility models (e.g., shorter movement periods). Lastly, PDRM can also be implemented as a prototype and evaluated more accurately in a testbed.

REFERENCES

AHLGREN, B. et al. A survey of information-centric networking. **IEEE Communications Magazine**, v. 50, n. 7, p. 26–36, July 2012. ISSN 0163-6804.

ANASTASIADES, C.; BRAUN, T.; SIRIS, V. A. Information-Centric Networking in Mobile and Opportunistic Networks. In: GANCHEV, I.; CURADO, M.; KASSLER, A. (Ed.). **Wireless Networking for Moving Objects: Protocols, Architectures, Tools, Services and Applications**. Cham: Springer International Publishing, 2014. p. 14–30. ISBN 978-3-319-10834-6. Disponível em: <https://doi.org/10.1007/978-3-319-10834-6_2>. Acesso em: 21 nov. 2017.

ANDREWS, J. G. et al. What Will 5G Be? **IEEE Journal on Selected Areas in Communications**, v. 32, n. 6, p. 1065–1082, June 2014. ISSN 0733-8716.

BAYHAN, S. et al. On Content Indexing for Off-Path Caching in Information-Centric Networks. In: **Proceedings of the 3rd ACM Conference on Information-Centric Networking**. New York, NY, USA: ACM, 2016. (ACM-ICN ’16), p. 102–111. ISBN 978-1-4503-4467-8. Disponível em: <http://doi.acm.org/10.1145/2984356.2984372>. Acesso em: 21 nov. 2017.

CAROFIGLIO, G. **Cisco Announces Important Steps toward Adoption of Information-Centric Networking**. Cisco, 2017. Blog. Disponível em: <https://blogs.cisco.com/sp/cisco-announces-important-steps-toward-adoption-of-information-centric-networking>. Acesso em: 21 nov. 2017.

CISCO. **Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016–2021**. [S.l.], 2017.

COHEN, B. Incentives build robustness in BitTorrent. In: **Workshop on Economics of Peer-to-Peer systems**. [S.l.: s.n.], 2003. v. 6, p. 68–72.

ELAYOUBI, S.-E.; ROBERTS, J. Performance and Cost Effectiveness of Caching in Mobile Access Networks. In: **Proceedings of the 2Nd ACM Conference on Information-Centric Networking**. New York, NY, USA: ACM, 2015. (ACM-ICN ’15), p. 79–88. ISBN 978-1-4503-3855-4. Disponível em: <http://doi.acm.org/10.1145/2810156.2810168>. Acesso em: 21 nov. 2017.

FAYAZBAKHSH, S. K. et al. Less Pain, Most of the Gain: Incrementally Deployable ICN. **SIGCOMM Comput. Commun. Rev.**, ACM, New York, NY, USA, v. 43, n. 4, p. 147–158, ago. 2013. ISSN 0146-4833. Disponível em: <http://doi.acm.org/10.1145/2534169.2486023>. Acesso em: 21 nov. 2017.

GARCIA-LUNA-ACEVES, J.; MIRZAZAD-BARIJOUGH, M.; HEMMATI, E. Content-Centric Networking at Internet Scale Through The Integration of Name Resolution and Routing. In: **Proceedings of the 3rd ACM Conference on Information-Centric Networking**. New York, NY, USA: ACM, 2016. (ACM-ICN ’16), p. 83–92. ISBN 978-1-4503-4467-8. Disponível em: <http://doi.acm.org/10.1145/2984356.2984359>. Acesso em: 21 nov. 2017.

GASTI, P. et al. DoS and DDoS in Named Data Networking. In: **2013 22nd International Conference on Computer Communication and Networks (ICCCN)**. [S.l.: s.n.], 2013. p. 1–7. ISSN 1095-2055.

GRASSI, G. et al. VANET via Named Data Networking. In: **2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)**. [S.l.: s.n.], 2014. p. 410–415.

ITU, I. T. U. **New ITU Standard: Information-Centric Networking to Support 5G**. ITU, 2017. Blog. Disponível em: <http://newslog.itu.int/archives/1527>. Acesso em: 21 nov. 2017.

JACOBSON, V. et al. Custodian-based information sharing. **IEEE Communications Magazine**, v. 50, n. 7, p. 38–43, July 2012. ISSN 0163-6804.

JACOBSON, V. et al. Networking Named Content. In: **Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies**. New York, NY, USA: ACM, 2009. (CoNEXT '09), p. 1–12. ISBN 978-1-60558-636-6. Disponível em: <http://doi.acm.org/10.1145/1658939.1658941>. Acesso em: 21 nov. 2017.

JOHNSON, D. D. B.; ARKKO, J.; PERKINS, C. E. **Mobility Support in IPv6**. RFC Editor, 2015. RFC 6275. (Request for Comments, 6275). Disponível em: <https://rfc-editor.org/rfc/rfc6275.txt>. Acesso em: 21 nov. 2017.

KO, H. et al. A proactive content pushing scheme for provider mobility support in information centric networks. In: **2014 IEEE 11th Consumer Communications and Networking Conference (CCNC)**. [S.l.: s.n.], 2014. p. 523–524. ISSN 2331-9852.

KUTSCHER, D. et al. **Information-Centric Networking (ICN) Research Challenges**. RFC Editor, 2016. RFC 7927. (Request for Comments, 7927). Disponível em: <https://rfc-editor.org/rfc/rfc7927.txt>. Acesso em: 21 nov. 2017.

LAOUTARIS, N.; CHE, H.; STAVRAKAKIS, I. The LCD interconnection of LRU caches and its analysis. **Performance Evaluation**, v. 63, n. 7, p. 609 – 634, 2006. ISSN 0166-5316. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0166531605000611>. Acesso em: 21 nov. 2017.

LEHMANN, M. B.; BARCELLOS, M. P.; MAUTHE, A. Providing Producer Mobility Support in NDN through Proactive Data Replication. In: **NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium**. [S.l.: s.n.], 2016. p. 383–391.

MAJEED, M. F. et al. Pre-caching: A proactive scheme for caching video traffic in named data mesh networks. **Journal of Network and Computer Applications**, v. 87, n. Supplement C, p. 116 – 130, 2017. ISSN 1084-8045. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1084804517301145>. Acesso em: 21 nov. 2017.

MASTORAKIS, S. et al. ndnSIM 2.0: A new version of the NDN simulator for NS-3. **NDN, Technical Report NDN-0028**, 2015.

MAURI, G. et al. Optimal Content Prefetching in NDN Vehicle-to-Infrastructure Scenario. **IEEE Transactions on Vehicular Technology**, v. 66, n. 3, p. 2513–2525, March 2017. ISSN 0018-9545.

MICK, T.; TOURANI, R.; MISRA, S. MuNCC: Multi-hop Neighborhood Collaborative Caching in Information Centric Networks. In: **Proceedings of the 3rd ACM Conference on Information-Centric Networking**. New York, NY, USA: ACM, 2016. (ACM-ICN '16), p. 93–101. ISBN 978-1-4503-4467-8. Disponível em: <http://doi.acm.org/10.1145/2984356.2984375>. Acesso em: 21 nov. 2017.

MOSKO, M. et al. **CCNx 1.0 Protocol Architecture**. [S.l.], 2015.

NSF/INTEL. **NSF/Intel Partnership on Information-Centric Networking in Wireless Edge Networks (ICN-WEN)**. NSF, 2016. Program Solicitation. Disponível em: <https://www.nsf.gov/pubs/2016/nsf16586/nsf16586.pdf>. Acesso em: 21 nov. 2017.

PERKINS, C. E. **IP Mobility Support for IPv4, Revised**. RFC Editor, 2010. RFC 5944. (Request for Comments, 5944). Disponível em: <https://rfc-editor.org/rfc/rfc5944.txt>. Acesso em: 21 nov. 2017.

POLAKOS, P. ICN: Information Centric Networking A new networking model for mobile networks and beyond. In: **Proceedings of the Workshop on Information Centric Networking for 5G (IC5G 2016)**. [s.n.], 2016. Disponível em: <https://www.cisco.com/c/dam/m/en_us/service-provider/ciscoknowledgenetwork/files/601_06_29-16-ICN_29Jun2016_CKN_Final.pdf>. Acesso em: 21 nov. 2017.

PSARAS, I.; CHAI, W. K.; PAVLOU, G. Probabilistic In-network Caching for Information-centric Networks. In: **Proceedings of the Second Edition of the ICN Workshop on Information-centric Networking**. New York, NY, USA: ACM, 2012. (ICN '12), p. 55–60. ISBN 978-1-4503-1479-4. Disponível em: <http://doi.acm.org/10.1145/2342488.2342501>. Acesso em: 21 nov. 2017.

SAHA, S.; LUKYANENKO, A.; YLä-JääSKI, A. Cooperative caching through routing control in information-centric networks. In: **INFOCOM, 2013 Proceedings IEEE**. [S.l.: s.n.], 2013. p. 100–104. ISSN 0743-166X.

SAROIU, S. et al. An Analysis of Internet Content Delivery Systems. **SIGOPS Oper. Syst. Rev.**, ACM, New York, NY, USA, v. 36, n. SI, p. 315–327, dez. 2002. ISSN 0163-5980. Disponível em: <http://doi.acm.org/10.1145/844128.844158>. Acesso em: 21 nov. 2017.

SHARMA, A.; VENKATARAMANI, A.; SITARAMAN, R. K. Distributing Content Simplifies ISP Traffic Engineering. **SIGMETRICS Perform. Eval. Rev.**, ACM, New York, NY, USA, v. 41, n. 1, p. 229–242, jun. 2013. ISSN 0163-5999. Disponível em: <http://doi.acm.org/10.1145/2494232.2465764>. Acesso em: 21 nov. 2017.

SILVA, V. B. C. da; CAMPISTA, M. E. M.; COSTA, L. H. M. K. TraC: A Trajectory-aware Content distribution strategy for vehicular networks. **Vehicular Communications**, v. 5, n. Supplement C, p. 18 – 34, 2016. ISSN 2214-2096. Disponível em: <http://www.sciencedirect.com/science/article/pii/S221420961630033X>. Acesso em: 21 nov. 2017.

TUDUCE, C.; GROSS, T. A mobility model based on WLAN traces and its validation. In: **Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.** [S.l.: s.n.], 2005. v. 1, p. 664–674 vol. 1. ISSN 0743-166X.

TYSON, G. et al. A Survey of Mobility in Information-centric Networks. **Commun. ACM**, ACM, New York, NY, USA, v. 56, n. 12, p. 90–98, dez. 2013. ISSN 0001-0782. Disponível em: <http://doi.acm.org/10.1145/2500501>. Acesso em: 21 nov. 2017.

VASILAKOS, X. et al. Proactive Selective Neighbor Caching for Enhancing Mobility Support in Information-centric Networks. In: **Proceedings of the Second Edition of the ICN Workshop on Information-centric Networking**. New York, NY, USA: ACM, 2012. (ICN ’12), p. 61–66. ISBN 978-1-4503-1479-4. Disponível em: <http://doi.acm.org/10.1145/2342488.2342502>. Acesso em: 21 nov. 2017.

VIRGILIO, M.; MARCHETTO, G.; SISTO, R. PIT Overload Analysis in Content Centric Networks. In: **Proceedings of the 3rd ACM SIGCOMM Workshop on Information-centric Networking**. New York, NY, USA: ACM, 2013. (ICN ’13), p. 67–72. ISBN 978-1-4503-2179-2. Disponível em: <http://doi.acm.org/10.1145/2491224.2491225>. Acesso em: 21 nov. 2017.

WANG, J. M.; ZHANG, J.; BENSAOU, B. Intra-AS Cooperative Caching for Content-centric Networks. In: **Proceedings of the 3rd ACM SIGCOMM Workshop on Information-centric Networking**. New York, NY, USA: ACM, 2013. (ICN ’13), p. 61–66. ISBN 978-1-4503-2179-2. Disponível em: <http://doi.acm.org/10.1145/2491224.2491234>. Acesso em: 21 nov. 2017.

WOO, T. et al. Proactive neighbor pushing for enhancing provider mobility support in content-centric networking. In: **2014 Sixth International Conference on Ubiquitous and Future Networks (ICUFN)**. [S.l.: s.n.], 2014. p. 158–163. ISSN 2165-8528.

WäHLISCH, M.; SCHMIDT, T. C.; VAHLENKAMP, M. Backscatter from the data plane – Threats to stability and security in information-centric network infrastructure. **Computer Networks**, v. 57, n. 16, p. 3192 – 3206, 2013. ISSN 1389-1286. Information Centric Networking. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1389128613002259>. Acesso em: 21 nov. 2017.

ZHANG, Y. et al. A survey of mobility support in Named Data Networking. In: **2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)**. [S.l.: s.n.], 2016. p. 83–88.

APPENDIX A — RESUMO EXPANDIDO

Neste capítulo, apresenta-se um resumo expandido da tese de doutorado. Primeiramente, será feita uma introdução e contextualização sobre mobilidade de conteúdo. Em seguida, será apresentada uma visão geral da proposta, denominada *Proactive Data Replication Mechanism* (PDRM). Por fim, serão discutidos os principais resultados obtidos através da avaliação.

Introdução

O tráfego móvel tem aumentado ano a ano e representa uma parcela considerável do tráfego de rede. Ao final de 2016, havia 8 bilhões de dispositivos móveis que geraram 7,2 exabytes de tráfego por mês (CISCO, 2017). Esses valores representam um aumento de quase meio bilhão (429 milhões) de dispositivos e 63% do tráfego mensal em relação ao ano anterior (CISCO, 2017). Mesmo com essa tendência crescente, a arquitetura da Internet de hoje não consegue fornecer suporte adequado para a mobilidade de usuários apesar de mecanismos como *Mobile IPv4* (PERKINS, 2010) e *Mobile IPv6* (JOHNSON; ARKKO; PERKINS, 2015). A causa é uma incompatibilidade entre a arquitetura da Internet e as demandas em constante evolução dos usuários.

Recentemente, novas propostas foram discutidas para mudar significativamente a arquitetura da Internet. Uma das propostas mais promissoras para atender às demandas dos usuários, incluindo o suporte à mobilidade de usuários, é a Rede Centrada em Informação (*Information-Centric Networking*, ICN) (AHLGREN et al., 2012). Este paradigma de rede muda de uma abordagem centrada no host (Internet atual) para uma centrada no conteúdo. A proposta de arquitetura de rede mais proeminente no modelo ICN é a *Named Data Networking* (NDN) (JACOBSON et al., 2009).

NDN é uma arquitetura de rede proposta para solucionar as deficiências existentes da Internet atual, incluindo mobilidade. Inspirado pelo comportamento predominante dos usuários (ou seja, o interesse dos usuários em conteúdo em vez de fontes ou mecanismos de entrega), NDN concentra-se no conteúdo, em vez dos *hosts*, e cria a arquitetura de rede em torno dele. O desenvolvimento do NDN ganhou uma força considerável com o apoio da indústria (CAROFIGLIO, 2017; POLAKOS, 2016; ITU, 2017; NSF/INTEL, 2016) e sua contribuição para o trabalho do *IRTF ICN Research Group*[1]. No geral, NDN

---

[1] <https://irtf.org/icnrg>

mostrou ser uma solução promissora para lidar com a tendência crescente de disseminação de conteúdo e mobilidade (CISCO, 2017) e pare ser o componente central do suporte à mobilidade na pesquisa e desenvolvimento de redes 5G (ANDREWS et al., 2014).

A arquitetura NDN é projetada para ajudar os produtores a disseminar e manter seu conteúdo disponível. Três recursos destacam-se para suportar a mobilidade de conteúdo: o conteúdo em NDN é replicável em toda a rede, as caches na rede podem aumentar o número de cópias de conteúdo e qualquer dispositivo que possua uma cópia do conteúdo pode satisfazer os pedidos recebidos. Como conseqüência, o conteúdo pode ter *provedores* adicionais além do produtor (ou *host* principal) para sua disseminação.

Por um lado, o design centrado no conteúdo permite que a arquitetura NDN ofereça suporte nativo para a mobilidade de consumidor. O modelo de comunicação empregado é dirigido ao destinatário e sem conexão, o que permite que consumidores retomem a recuperação do conteúdo após o movimento, apenas re-enviando suas requisições de forma transparente. Esse suporte é mais simples e eficiente do que *MobileIPv6* (JOHNSON; ARKKO; PERKINS, 2015). Por outro lado, o design do NDN não é suficiente para lidar com *mobilidade de conteúdo*, oferecendo apenas suporte limitado e exigindo soluções de rede mais complexas para alcançar esse objetivo plenamente (TYSON et al., 2013; KUTSCHER et al., 2016; ZHANG et al., 2016).

O desafio da mobilidade de conteúdo em comparação com a mobilidade de consumidor é que os objetos devem ser mantidos disponíveis e acessíveis para os consumidores apesar de um possível movimento ou indisponibilidade de seus provedores (ANASTASIADES; BRAUN; SIRIS, 2014). Com a possibilidade de vários provedores para o mesmo conteúdo em NDN, o conteúdo está disponível se houver pelo menos um provedor de conteúdo com uma cópia dele na rede. Assim, NDN assegura a mobilidade de conteúdo quando os consumidores podem continuar a recuperar algum conteúdo, apesar da indisponibilidade de uma fração de seus provedores (produtor incluído) devido ao movimento. Dado o problema apresentado, a seguinte pergunta guia esta tese.

Como melhorar o suporte à mobilidade de conteúdo em NDN usando uma abordagem que tanto beneficiará quanto potencializará as características centradas em conteúdo da arquitetura NDN?

Esta tese investiga como usar replicação de dados pró-ativa e com consciência de localização para melhorar o suporte à mobilidade de conteúdo em NDN. A replicação de conteúdo pró-ativa é uma abordagem sub-explorada para o suporte à mobilidade de conteúdo em NDN e este trabalho estuda formas de implementá-lo de forma a se beneficiar ao máximo da arquitetura NDN. Como resultado, é proposto o Proactive Data Replication Mechanism (PDRM), composto por duas operações: *Descoberta da Vizinhança* e *Envio de Conteúdo*.

O objetivo do PDRM é aumentar a disponibilidade de conteúdo através da redundância de dados com o uso eficiente da rede e dos recursos dos usuários finais. PDRM é um serviço opcional para produtores móveis replicarem o conteúdo de forma pró-ativa. Ao contrário das abordagens anteriores, um produtor móvel que usa PDRM aprende sobre sua vizinhança e usa essa informação para influenciar as decisões de replicação. O passo de aprendizagem é a principal característica que permite ao PDRM otimizar o processo de replicação de conteúdo para ter um alto desempenho de disseminação e baixo consumo de recursos.

O design do PDRM suporta a mobilidade de conteúdo da seguinte forma. A replicação pró-ativa permite que novas cópias de objetos sejam criadas e divulgadas antes que os consumidores solicitem o conteúdo, o que aumenta o uso de recursos do NDN em uma distribuição de conteúdo subseqüente. Estar ciente do contexto de localização permite que o produtor de conteúdo colete informações sobre seus objetos (por exemplo, popularidade) e outros usuários nas proximidades. Este conhecimento pode ser usado para inferir quais objetos tendem a se tornar populares e se os usuários finais estão dispostos a se tornar provedores temporários de tal conteúdo. O armazenamento extra nos usuários finais ajuda a rede de forma oportunista a aumentar a disponibilidade de conteúdo e melhorar a sua disseminação.

É medido o desempenho e sobrecarga do PDRM sob vários aspectos. Na avaliação, PDRM e propostas do estado-da-arte sao implementadas no ndnSIM (MASTORAKIS et al., 2015) para comparação. Na avaliação, é medido o impacto do número de provedores disponíveis e da capacidade de cache da rede no desempenho do PDRM. Por fim, é comparado PDRM com NDN padrão e propostas do estado-da-arte, *Data Spot* (WOO et al., 2014) e *Data Depot* (JACOBSON et al., 2012), quando há períodos de indisponibilidade do produtor.

As principais contribuições da tese são resumidos da seguinte forma.

- Uma investigação detalhada da replicação de dados pró-ativa e consciente de localização para suportar a mobilidade de conteúdo em NDN.
- A proposta do Proactive Data Replication Mechanism (PDRM), cujo principal diferencial é aprender sobre a vizinhança para melhorar o suporte à mobilidade de conteúdo em NDN.

- A avaliação do desempenho do PDRM em diferentes configurações de cenários e comparativamente ao NDN padrão e duas propostas do estado-da-arte (Data Depot e Data Spot).

PDRM

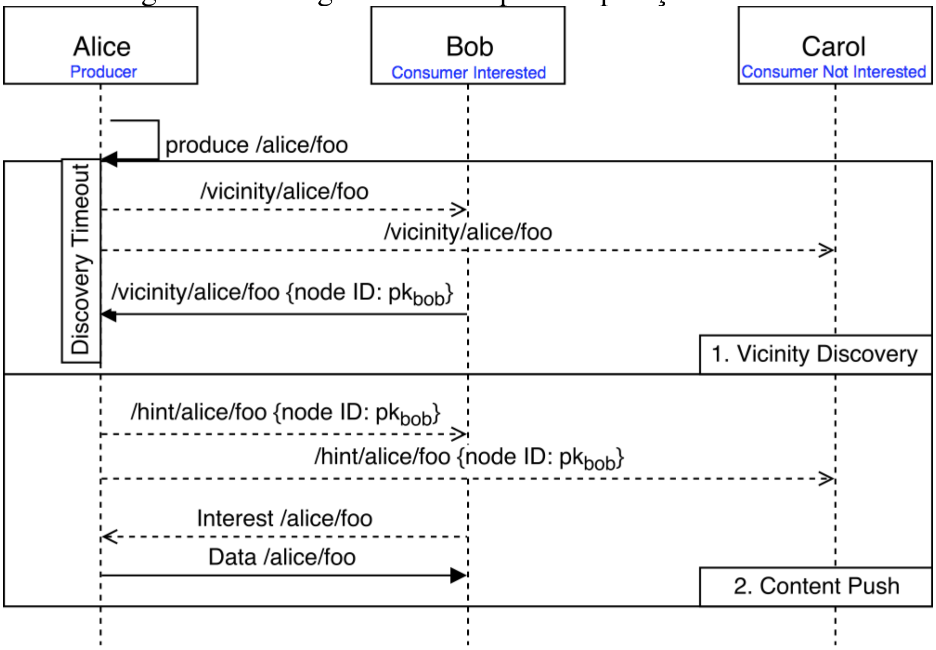
Nesta seção, é apresentada uma visão geral do *Proactive Data Replication Mechanism* (PDRM), proposta para melhorar o suporte à mobilidade de conteúdo em NDN. PDRM reduz o impacto da indisponibilidade do produtor na disponibilidade do conteúdo através da redundância de dados, com o uso eficiente de recursos disponíveis da rede e dos usuários. Depois que um produtor cria um objeto, ele pode empurrar cópias para outros usuários, que se tornam provedores. Os resultados esperados são um aumento na taxa de obtenção de conteúdo e, conseqüentemente, uma melhoria na QoE do consumidor, particularmente seu tempo de download, com sobrecarga limitada para os usuários e a rede.

PDRM é um mecanismo de replicação pró-ativo, com consciência de localização, de melhor esforço e baseado em dicas que explora os recursos disponíveis dos usuários finais na vizinhança. Ele é composto de duas operações: descoberta da vizinhança e envio de conteúdo. Para lidar de forma pró-ativa com a mobilidade, um produtor pode executá-las nos objetos de escolha uma ou mais vezes após sua criação. Primeiro, o produtor envia uma sonda para saber mais sobre os dispositivos próximos e a popularidade do conteúdo neles. Então, o produtor decide, com base na informação coletada, se deve replicar os objetos de escolha usando a operação de envio de conteúdo.

A execução do PDRM é ilustrada com um exemplo composto por Alice (produtora), Bob e Carol (usuários na vizinhança de Alice). Figura A.1 apresenta a troca de mensagens da execução do PDRM. Este exemplo começa com Alice criando o objeto chamado </alice/foo> e iniciando uma descoberta de vizinhança para ele. Alice transmite uma sonda a sua vizinhança, que contém Bob (interessado no conteúdo) e Carol (não interessado no conteúdo). Bob responde à sonda recebida com sua chave pública, enquanto Carol simplesmente descarta a mensagem.

Depois de algum período para coletar as respostas, Alice conclui a descoberta de vizinhança e usa o conhecimento reunido para replicar o objeto produzido. Neste exemplo, Alice verifica que só Bob está interessado em se tornar um provedor de </alice/foo> e envia-lhe uma dica para o conteúdo usando uma mensagem de *broadcast* com a

Figure A.1: Diagrama de Tempo da Operação do PDRM



chave pública dele. Quando Bob recebe a sugestão, ele procede a solicitar o objeto usando o protocolo NDN padrão. Carol também recebe a dica, mas a descarta porque não está destinada a ela.

Principais Resultados

Ao longo da tese, PDRM é avaliado em diferentes aspectos. Primeiramente, foi avaliado o impacto do número de provedores disponíveis e da capacidade de cache na rede no desempenho do PDRM. As conclusões encontradas são que PDRM melhora gradualmente o desempenho do download ao adicionar os primeiros provedores de conteúdo extra até convergir os resultados quando o número de provedores satura. Neste ponto, que varia de acordo com a configuração da rede, PDRM pode até mesmo deixar de obter o melhor desempenho devido à replicação excessiva de objetos, que potencialmente polui as caches com conteúdo impopular. Em relação à capacidade de cache na rede, é mostrado que PDRM funciona inversamente proporcional às caches, oferecendo desempenho constante independente da cache disponível.

Em seguida, a avaliação do PDRM é concluída com uma comparação entre PDRM, NDN padrão e duas propostas do estado-da-arte, *Data Depot* e *Data Spot*, em dois cenários com períodos de indisponibilidade do produtor. O primeiro tem conteúdo uniformemente distribuído na rede e representa um limite inferior para o desempenho. PDRM reduz os tempos de download entre 23.13% (Vanilla NDN e Data Spot) e 6.04% (Data Depot), diminui a carga do produtor para 65,53% e gera uma sobrecarga 80% inferior ao do Data Depot. O segundo cenário usa conteúdo agrupado em torno de áreas de interesse, representando um limite superior de desempenho. PDRM evidencia todo o seu potencial neste cenário. Comparado com o NDN padrão e Data Spot, PDRM reduz os tempos de download até 61,09%, carga do produtor até 76,26% e tráfego inter-domínio até 46,50%. Como esperado, o caso da *Content Delivery Network* (Data Depot com um armazenamento para o catálogo completo) supera os mecanismos restantes em ambos os cenários. No primeiro cenário, PDRM tem resultados semelhantes ao Data Depot armazenando 200 objetos, enquanto no segundo 700 objetos, mas utiliza significativamente menos recursos e gera menor sobrecarga.

Em geral, a avaliação indica resultados promissores para a replicação pró-ativa no contexto de mobilidade, apesar dessa abordagem não ser sugerida em redes regulares sem mobilidade (SHARMA; VENKATARAMANI; SITARAMAN, 2013; FAYAZBAKHSH et al., 2013). Quanto a trabalhos futuros, existem três direções de extensão: refinamento, avaliação e implantação. PDRM pode ser refinado para inferir melhor a popularidade do conteúdo e como lidar com ela para evitar a replicação excessiva, bem como sua economia e como operar com vários produtores concorrentes para os recursos dos provedores. A segunda direção considera como continuar a avaliação do PDRM: cenários diferentes, como aplicações em tempo real ou de *streaming*, e modelos de mobilidade (por exemplo, períodos de movimento mais curtos). Por fim, o PDRM também pode ser implementado como um protótipo e avaliado com mais precisão em um *testbed*.

APPENDIX B — PAPER AT IEEE/IFIP NOMS 2016

Title: Providing Producer Mobility Support in NDN Through Proactive Data Replication

Conference: IEEE/IFIP Network Operations and Management Symposium (NOMS 2016)

Type: Main track (full-paper)

Qualis: A2

Date: April 25-29, 2016

Location: Istanbul, Turkey

Digital Object Identifier (DOI): <10.1109/NOMS.2016.7502835>

Abstract: Named Data Networking (NDN) is a novel architecture expected to overcome limitations of the current Internet. User mobility is one of the most relevant limitations to be addressed. NDN supports *consumer* mobility by design, but fails to offer the same level of support for *producer* mobility. Existing approaches to extend NDN are host-centric, which conflicts with NDN principles, and provide limited support for producer mobility. This paper proposes a content-centric strategy that replicates and pushes objects proactively, and unlike previous approaches, takes full advantage of NDN routing and caching features. We compare the proposed strategy with default NDN mechanisms regarding content availability, consumer performance, and network overhead. The evaluation results indicate that our strategy can increase the hit rate of objects by at least 46% and reduce their retrieval time by over 60%, while not adding significant overhead.

Providing Producer Mobility Support in NDN Through Proactive Data Replication

Matheus B. Lehmann, Marinho P. Barcellos
Institute of Informatics
Federal University of Rio Grande do Sul, Brazil
Email: {mblehmann, marinho}@inf.ufrgs.br

Andreas Mauthe
School of Computing and Communications
Lancaster University, UK
&
KOM, TU Darmstadt, Germany
Email: a.mauthe@lancaster.ac.uk

***Abstract*—Named Data Networking (NDN) is a novel architecture expected to overcome limitations of the current Internet. User mobility is one of the most relevant limitations to be addressed. NDN supports *consumer* mobility by design but fails to offer the same level of support for *producer* mobility. Existing approaches to extend NDN are host-centric, which conflicts with NDN principles, and provide limited support for producer mobility. This paper proposes a content-centric strategy that replicates and pushes objects proactively, and unlike previous approaches, takes full advantage of NDN routing and caching features. We compare the proposed strategy with default NDN mechanisms regarding content availability, consumer performance, and network overhead. The evaluation results indicate that our strategy can increase the hit rate of objects by at least 46% and reduce their retrieval time by over 60%, while not adding significant overhead.**

I. INTRODUCTION

Mobile traffic has been increasing yearly and has now become a central part of computer networks. In 2014, 7.4 billion mobile devices increased their monthly traffic by 69% compared to 2013, generating 2.5 exabytes per month [1]. Despite this growing trend, today's Internet architecture fails to provide an adequate user mobility support with mechanisms such as Mobile IPv4 [2] and Mobile IPv6 [3].

Named Data Networking (NDN) [4], a network architecture proposal, aims to address some of the Internet current shortcomings, including support for user mobility. There are three key elements in NDN: content, users, and routers. Content is comprised of objects, which are single pieces of information. Users are entities that can produce, provide or consume objects. Routers are (static) devices which route and forward packets. They do not consume nor produce content, but can cache objects and provide them to interested consumers.

User mobility represents a challenge because users are associated with a device, which has limited resources (e.g. storage, bandwidth, battery), and which can be unavailable for a period of time before reappearing somewhere else. NDN natively supports *consumer* mobility: when mobile consumers move on an NDN network, they do not need to restore connections, sessions or update their locations. Instead, it is enough to re-send interest requests for non-retrieved data to fetch the desired content objects, which is possible because objects are idempotent, and NDN does not rely on end-to-end communication. In contrast, NDN does not support *producer* mobility. The difference compared to the consumer mobility is that the producer has to maintain the content objects available and provide them to requesters in spite of its possible movement or unavailability [5].

Producer mobility can be divided into two distinct periods: unavailability and re-attachment. While the former is characterized by the producer's lack of network connectivity during movement, the latter refers to the process of rejoining the network and restoring producer connectivity. The unavailability period is a more relevant research challenge, for it lacks proper support in current NDN. The latter period, in contrast, can be reasonably addressed through announcement messages.

In this paper, we focus on supporting producer mobility in NDN by keeping the content available. We propose a strategy based on a straightforward principle: proactively and efficiently replicating content by pushing it according to a placement policy. Its objective is to reduce the negative effects of producer mobility into content availability, without incurring significant performance loss for consumers nor overhead to the network. Unlike previous proposals, our strategy follows NDN principles and leverages its key features to overcome the loss of content availability induced by producer mobility.

The contributions of this paper are twofold. First, we propose a strategy to handle producer mobility in NDN based on data replication and aligned with NDN core principles. The strategy is able to keep content available despite producer mobility, with limited overhead. Second, we perform a detailed evaluation to study in which conditions data replication improves producer mobility support in NDN. The analysis focuses on understanding the impact of three key parameters of the strategy: vicinity size (producers knowledge about the network), replication degree, and placement policy. We measure their effects in two scenarios: a simpler case with a single producer, to highlight the effect of each strategy parameter; and one with multiple producers, to measure the effects of widely using the strategy, looking at the impact of the producers' local decision on the global results achieved.

The rest of the paper is organized as follows. Section II

978-1-5090-0223-8/16/$31.00 © 2016 IEEE

presents the state-of-the-art in NDN producer mobility. Section III describes the proposed strategy to handle producer mobility via data replication. Section IV presents the methodology used in our work to evaluate the proposed strategy. Section V discusses the results and main findings. Section VI concludes this paper and discusses future work.

II. STATE-OF-THE-ART

In this section, we present and discuss the most recent advances in supporting the producer unavailability period in NDN. First, we define more precisely what is the unavailability period and discuss ways a network architecture can address it. Then, we organize existing proposals into four categories and discuss each one of them.

A. *Unavailability Period*

The unavailability period occurs when the producer is unable to provide its content. A network architecture can address this in three ways: provide no support, reduce the producer unavailability through host-centric mechanisms, or keep the content available through content-centric ones. These three alternatives are discussed next.

The simplest approach is not to provide any extra producer unavailability support. In other words, the network architecture does not have any feature to aid the producer in keeping itself or its content available during this period. This lack of producer mobility support is detrimental to applications that require the producer (or its content) to be accessible most of the time. Nevertheless, this approach can be acceptable in scenarios where the producers move but remain connected.

The second way focuses on reducing the producer unavailability via host-based mechanisms. The network architecture can use seamless mobility or connection restore. The former aims at minimizing the unavailability during a hand-off. The latter stores the communication when the producer is unavailable and restores it once the producer rejoins the network.

The third and last way, content-centric, keeps the content available despite the producer mobility. A network architecture usually employs data replication and caching to fulfill this goal. This kind of approach is seen in Content Delivery Networks (CDN), Distributed Hash Tables (DHTs), and NDN. This approach enables other network elements besides the producer to provide the requested data.

B. *NDN Proposals*

The NDN architecture does not address producer unavailability by design. Recent proposals to extend it can be separated into the following categories: (*i*) proactive content push, (*ii*) store and forward requests, (*iii*) use of NDN default or extended support, and (*iv*) use of non-NDN techniques.

Proactive push. [6] aims at maintaining the content available through proactive replication, instead of keeping the producer available at all times. It focuses on the case that the producer moves during a data transfer. Prior to moving, the producer pushes data proactively towards the requester. The router that receives the data stores it, enabling future requests to be satisfied on behalf of the producer.

Store and forward requests. This category focuses on avoiding the loss of requests and the need for consumers to re-issue them. [7]–[11] propose the addition of a network element responsible for storing requests when the producer is unavailable and forward them once the producer returns. The difference between them is how requests are forwarded when the producer returns: updating the FIB tables [8], using an indirection point [7], existing NDN features or a combination of them [9]–[11].

Using NDN default communication or extending its messaging protocol. [12], [13] map persistent and temporary data names of mobile producers. The producer updates its binding information through the existing NDN messages. [14] uses the NDN messaging protocol to notify the network when it detects a degradation in the current link signal caused by movement. The notification allows the routers to react and maintain the reachability toward the producer. [15] proposes Kite, a scheme that uses routable anchors to track the producer movement. It extends the NDN protocol, using PIT entries to create breadcrumbs from the anchor to the producer through the use of traceable interest packets. [16] proposes a solution based on name resolution that extends the interest packets to contain a hint of where the content might be located.

Non-NDN techniques for mobility support. [17] uses greedy routing, which can coexist with NDN default routing protocol, combined with indirection points. [18] proposes Auspice, a global name service to provide a low lookup latency, small update cost, and high availability. Despite not focusing on NDN, the solution can be applied to name-based communication such as NDN. [19] combines Software Defined Networking (SDN) with NDN to perform global and local FIB updates. It reduces the cost of routing information updates by limiting its scope when handling mobility.

Unlike previous work, host-centric, the proposed strategy supports producer mobility in a content-centric fashion by increasing the availability of its content in the network. Although [6] refers to a similar idea, it is a very limited study. Related proposals provide limited support for producer mobility to date and their designs conflict with NDN principles [4].

III. DATA REPLICATION STRATEGY

In this section, we present our proposed strategy to address producer mobility in NDN through data replication. The overall objective of the proposal is to increase the content availability and minimize the impact of the producer unavailability. It is based on a straightforward principle, i.e. proactive replication of content by the producer. When a producer creates a content object, it may push one or more replicas to other users. The strategy has five aspects, to be discussed: (*i*) vicinity, (*ii*) content push operation, (*iii*) data replication degree, (*iv*) content placement policy, and (*v*) producer re-attachment.

A. *Vicinity*

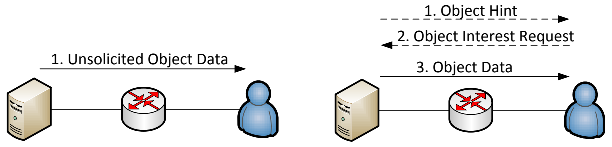
The vicinity of a device is defined as the set of nodes whose distances from the device are less than or equal to a threshold.

It can vary from the direct neighbors to the complete network. In the current NDN, a user device knows only about the router to which it is connected, which is the same as having threshold one.

We argue that a producer can benefit from a larger view and information about other devices. The proposed strategy extends NDN by expanding the device view to a vicinity, whose size varies with the topology and the device threshold. Useful information can be the availability of other devices, the interest on the content object, or the number of existing replicas in the vicinity to infer their popularity or rarity. Presently, we focus on the availability of devices. The producer attempts to obtain such information about devices in its vicinity, keeping it as soft state, and influencing the placement of content objects.

B. Content Push Operation

The concept of pushing data does not exist in the current NDN architecture, as the content dissemination in NDN is reactive rather than proactive (receiver-driven). There are two alternatives we could use to provide pushing in the NDN architecture: through unsolicited data or hints. They are exemplified in Figure 1.



(a) Unsolicited Data (b) Hints

Fig. 1. Pushing operation using unsolicited data or hints

In the first alternative (Figure 1(a)), the producer has to 'force' the content to some other devices. This operation could be achieved by changing the router protocol to add new packet types or semantics, allowing producers to send their content to any other device on the network at any given moment. However, this alternative goes against the NDN principle of receiver-driven communication and is less secure.

Therefore, we consider a second alternative (Figure 1(b)), in which producers suggest to other devices that they should request a given content object. A device that receives the indication may follow it or not. In the case of using the tip, the user requests the content and retrieves it from the producer. This alternative is implemented at the application layer, which makes it compatible with the NDN architecture design. Its downside is the number of exchanged messages (two interest and one data packets) and the time to complete the operation, which decreases the content dissemination performance.

C. Data Replication Degree

For each content object created, the producer decides how many copies to push, to improve availability. Considering the potentially high cost of this operation for both the producer and the network (e.g. a single replica will double the object transmission cost for the producer), we expect only small replication degrees (1-5) to be affordable and, even so, not applicable to all objects/producers.

Thus, to use network resources efficiently, the number of copies pushed is inversely proportional to the producer availability. In other words, a less available producer pushes more copies than a highly available one. A maximum replication degree parameter is used to define how aggressive the producer can push its data; it can take into consideration content popularity, rarity or available device resources.

D. Content Placement Policy

A producer uses its vicinity and replication degree to build a ranking of devices to receive a copy of an object. The ranking is built according to a set of metrics collected from the vicinity, such as device availability, interest in objects, movement patterns, and stability. For the moment, we consider two placement policies: *random* and *availability-based*. The former policy only requires knowledge about the vicinity, and dispenses with any specific information about the devices themselves, while the latter takes advantage of device availability information.

E. Producer Re-attachment

When a producer rejoins the network at a different location, its content has to be made available to other users. Specifically, the routers must update their routing information to route interest requests to the correct producer location. Currently, the strategy relies on the NDN's default mechanism to perform this operation, i.e. each provider re-announces itself at the new location, and routers propagate this information to converge their routing tables. This decision adds an announcement cost and a routing convergence time to the strategy performance.

IV. METHODOLOGY

The evaluation is based on a scenario with the following characteristics: a network in which every user is potentially mobile and a producer, users are not always available, content is user generated, and interest for content objects is location biased. This section describes the evaluation model, scenarios, parameters, and metrics.

A. Evaluation Model

The model describes an NDN network with mobility and serves as the basis to study the proposed strategy to support producer movement. Its fundamental elements are the devices and the content objects in the network. A device is characterized by a mobility pattern, interest in content objects, and available resources. Devices can be either users or routers. User devices are mobile elements that produce, provide, and consume content. Routers, static components that form the network infrastructure, route requests, forward data, and cache content objects to serve future requests.

A content object is a piece of information with some popularity that is produced and consumed by users. We assume that every object has the same size and is composed by one chunk. The life cycle of a content object consists of its creation, pushing, request, and caching, as discussed next.

A content object is **created** by a user (its producer), which can execute three actions: satisfy incoming interests for the object, push it to other devices or announce its availability. We assume that each user has a home network, which is pre-configured in the routers. As a result, the producer only announces an object creation when away from its home network.

After creating a content object, the producer can **push** it to other devices. The producer decides how many replicas are created and where they are placed in the network based on the strategy description in Section III-D. Each device receiving a copy of the object becomes a provider of this object, announces its possession, and can satisfy future interests for it. The routers in the path between the producer and a new provider store a copy in their cache and update their routing information for this content object with the new provider.

Whether an object was pushed or not, it may be sought by a consumer **request**. In the evaluation, we consider static consumers only, which means they do not move while receiving an object. When users request objects, they retrieve data from the closest provider, according to the routing information in routers. If a router in the path has the object in its cache, it will provide the data for the consumer instead of the provider. During the retrieval, the content object is cached in the routers in the path between the provider and the consumer. If not a single provider is available during the request, the retrieval fails, and the consumer re-issues a new interest request in the subsequent interval.

Throughout the lifetime of a content object, its copies may be **cached** by both users and routers. Users can retrieve any content object they desire but only become providers of the subset of objects that they decide to announce. Routers cache every data that goes through them. The cache of every device has a maximum size, which forces the device to replace old entries using LRU when caching an object in an already full cache. Further, a cache entry in a router has a maximum lifetime to emulate other traffic that goes through it.

B. Scenarios

We employ two scenarios in the evaluation, according to the number of objects/producers: one or multiple. In the first scenario, we analyze the benefits of the proposed strategy comparing it with default NDN. Further, a sensitivity analysis is performed to understand the impact of the primary parameters of the proposed strategy: vicinity size, replication degree, and placement policy.

The second analysis studies how our strategy works on a larger scale, with multiple content objects and producers. The strategy parameters used in this scenario are based on the results obtained from the previous analysis. The proposed strategy executes locally in each producer without a global view of the network regarding the available resources and object placement. Throughout this scenario evaluation, we investigate whether the strategy can achieve good global results solely based on local decisions of each producer.

C. Topology & Workload

The model is implemented as a discrete simulation[1]. The topology is based on a random geometric graph, which covers a square area. The topology is composed of 33 routers and 56 links between them with a homogeneous latency of 10ms. The topology has an average shortest path of around 3.83 hops, its eccentricity (the largest shortest path between any two vertices) is 10 hops, and the clustering coefficient (how grouped are the nodes) is 48%. The network has 750 user devices uniformly distributed, averaging around 23 users per router [20]. The simulation is executed over 240 time steps, each representing around 1 minute.

The mobility model parameters describe how users move through the network. The user behavior is described by active-inactive cycles [21], and its movement is modeled using the Graph-based Random Waypoint Model [22]. User behavior is a sequence of successive sessions, characterized by periods of connectivity (i.e. active session) and inactivity (i.e. movement). The session duration follows the distribution measured in [21], in which 75% of the sessions are up to 7 minutes and 92% are shorter than an hour. It is roughly represented by a Pareto distribution with a shape of 0.38 and scale of 0.18. We assume that the longest session lasts at most 80 time steps (or minutes). Users move in a subset of between 2 and 9 routers (possible locations) uniformly distributed in the network [23]. They move between their possible locations following the graph paths at a random speed between 1 and 5 time steps per hop [21].

The workload of the simulation is based on User-Generated Content (UGC) [24]. Either one or multiple content objects are produced according to the evaluation scenario. In the case of multiple objects, 250 content objects are created, which results in 20% of producers in the network. All objects of the catalog have the same size of 10MB and are formed by a single piece of data. Their popularity distribution follows a Zipf distribution with $\alpha = 0.44$. This results in a long-tail of unpopular objects, characteristic of User-Generated Content [25]. In the context of UGC, consumers are concentrated based on their geo-location. In the evaluation, 75% of consumers of an object are grouped in an area whereas the rest is randomly distributed in the network [18]. The cache size of each device stores 1% (default value for NDN) of the maximum catalog size, and in-network caches have a lifespan of 3 time steps to simulate other traffic on the network. Besides the data objects, there is also non-cacheable control objects for the vicinity learning and content announcement, whose size are 10KB each.

The vicinity parameters are varied between 1 (default NDN and without the strategy) and 11 hops (complete view of the network). The maximum replication degree varies between 1 and 6 replicas. The strategy selects either a random or the best available device to push content to.

---

[1] The source code, input, and output files are available in https://github.com/mblehmann/noms-2015

D. Metrics

We evaluate three aspects of the strategy: content availability, consumer performance, and overhead. The following metrics are used to evaluate them:

- Content Hit Rate: the percentage of successful retrievals of content by the consumers. A higher hit rate indicates that the content has a higher availability.
- Content Retrieval Time: the average time for consumers to retrieve a content object from the closest provider. If no copies are found, the time of a failure request attempt (i.e. the time to send a request to the producer) is added to the overall retrieval time. This metric also includes provider announcement time.
- Number of Packets: the network overhead. A higher number of packets results in more traffic in the network, which in turn can cause congestions and delays. We measure the number of packets required for vicinity learning, pushing, and announcement separately.
- Data Volume: how much bandwidth is consumed. This metric complements previous one to describe the overhead caused by our strategy. A higher data volume can saturate the network bandwidth usage, and hence decreases its overall performance. The data volume takes into consideration both the packet size and the distance traveled.

V. RESULTS

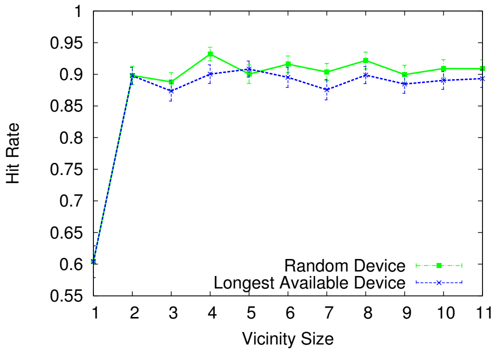
In this section, we first present and discuss the results for the single producer scenario, followed by the one with multiple mobile producers.

A. Single Producer Scenario

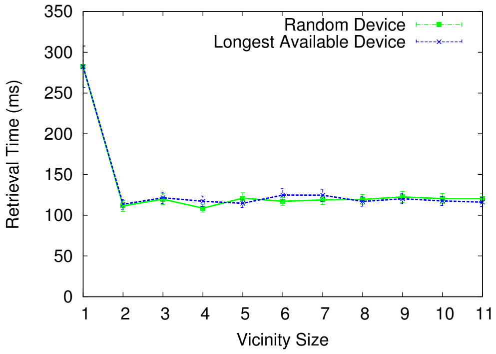
The goal of this analysis is first to quantify how well NDN supports producer mobility and compare it with the proposed strategy. In addition, it aims to assess the impact of the parameters vicinity size, placement policy, and replication degree on the strategy performance.

1) Vicinity Size: Figure 2 shows how the vicinity size and placement policies influence on content hit rate and retrieval time. When the vicinity size is set to one, the strategy is not executed because the producer knows only the router to which it is connected and does not know any device to push data to. Because of the topology properties, a vicinity of size eleven guarantees that the producer will have a full view of the network[2]. Regarding the placement policies, we evaluate two policies: random, and longest available device.

With the NDN's default settings, producer mobility causes the content to have a low availability. The content achieves a hit rate just over 60%, as shown in Figure 2(a). The consequence of the content unavailability also reflects on the consumer performance. When consumers are unable to retrieve the content, their request times out and they need to issue a new interest request. The results presented in Figure 2(b) show that consumers take about 282ms to retrieve an object due to this behavior.



(a) Hit Rate



(b) Retrieval Time

Fig. 2. Hit rate and retrieval time results varying the vicinity size

Expanding the vicinity to two generally allows the producer to discover other devices and push data to them. The results improve significantly compared to those with default NDN, as shown in Figure 2. By pushing one replica of the content object, the producer increases the content hit rate from 60% to around 89%. It also reflects on the consumer performance because it reduces the chance of not being able to find the desired object. As a matter of fact, the copy reduces the consumers retrieval time by 60% from 282ms down to 111ms.

Figure 2 also shows the effects of the placement policy variation. The placement policy based on availability takes into consideration the remaining time of the devices' current session to find those that will stay the longest in the short term. Regardless of the vicinity size, the impact of the availability-based placement policy was negligible in comparison to the random. Analyzing the results with vicinities whose threshold lies between two and eleven, we see that the results vary between 88% and 93% for hit rate, and 108ms and 122ms for retrieval time without a clear trend.

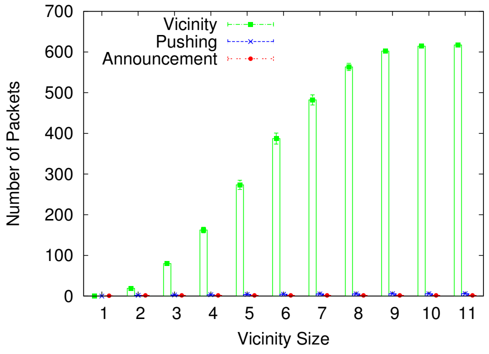
The producer's extra knowledge did not improve the results due to the combination of poor information quality and the network dynamics. For simplicity, the producer learns only about the current session of another device. Besides, the fact that session durations are independent, it is not possible to predict

---

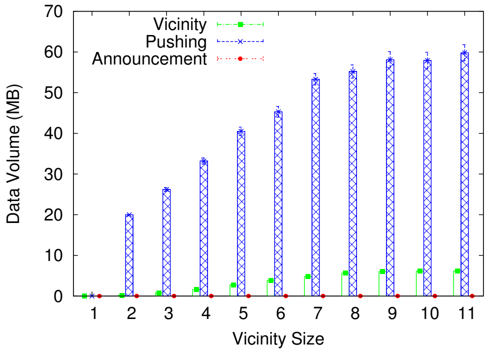
[2] Note, a producer with a vicinity of size nine or larger may have complete knowledge about the network due to its topological position.

the duration of future sessions of a given device. Hence, for the sake of the evaluation, the availability information of a device is not useful in the medium and long-term. So, in the remaining analysis, we present only the results for the random device policy. Note, however, that the proposed strategy allows a producer to leverage network properties and availability profile of devices to improve content placement.

The results of the network overhead caused by our strategy in terms of processing and bandwidth are shown in Figure 3. Figure 3(a) shows that a larger vicinity requires more packets to learn about other devices due to the higher number of known devices. The more packets created, the more processing a router requires to route and forward them, which may overload the network. Because of the network topology and its clustering coefficient of 48%, each expansion of the vicinity slightly increments the number of known devices. In comparison, a highly clustered topology would exponentially increase the number of vicinity packets sent in the first steps and then quickly converge to the maximum value. From a vicinity of size nine onwards a producer can have a complete view of the network, which explains why the number of packets for the vicinity learning converges at a vicinity of size nine. We also see that the number of packets for the pushing and announcement operations does not vary with different vicinity sizes.



(a) Number of Packets



(b) Data Volume

Fig. 3. Network overhead results varying the vicinity size

To have a better understanding of the actual network overhead, the data volume generated by the producer is also analyzed. The results are shown in Figure 3(b). The dominant aspect in the data volume is the pushing operation. The minimum value obtained for the data volume overhead is 20MB, which is achieved by pushing the 10MB-sized object to a device two hops away. As the vicinity size expands, the data overhead of pushing just one replica increases up to 60MB (six times the original object) because the producer may push a replica farther away, which consumes more resources. The data volume depends directly on the object size and the travel distance. Since the producer pushes the replica to a random device, the results reflect the average distance of the known devices in the vicinity. In the evaluation, the data volume growth is almost linear due to the network topology and uniform device distribution. If the producer uses a different placement policy (e.g. one that favors closer devices), the network has a different topology or devices have another distribution pattern the growth might not be linear. Learning about other devices in a small vicinity does not add significant data to the network, but as it is expanded the data volume overhead becomes a relevant factor. With a complete view of the network, the producer may generate an extra 6MB to learn about other devices.

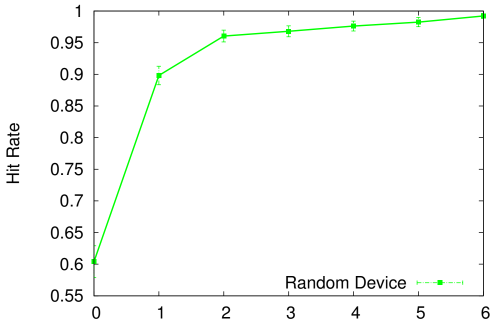
The results of combining the cost and benefits analysis show that a vicinity of size two is the best choice. The producer increases significantly the hit rate and reduces the consumer retrieval time by knowing at least one device to push a replica to. The small vicinity does not add significant overhead because it restricts the communication close to the producer, hence reducing the overall resource consumption of the network. In the remainder analysis, a vicinity size of two is assumed.

2) Replication Degree: Figure 4 shows the results for hit rate and retrieval time with a different number of maximum replicas pushed by the producer. As we can see in the results shown in Figure 4(a) when the producer pushes at least one copy it increases the content hit rate from 60% with default NDN to 89%. It reflects directly on the consumer performance presented in Figure 4(b), where the retrieval time is reduced by 60% from 282ms to 111ms.

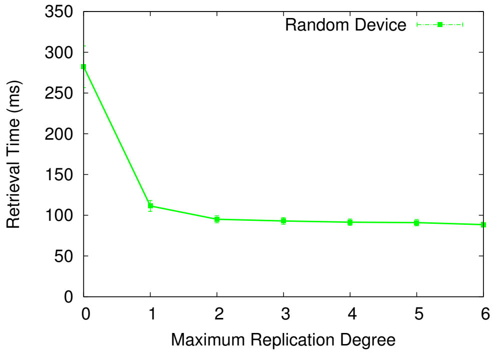
When the producer pushes more than one copy to other devices, it improves further the results. With two copies, the content hit rate increases from 88% to around 96%. With three or more copies, the hit rate gain is marginal until it converges to 99% with six copies. This behavior is also seen in the consumer performance. Pushing two copies reduces the retrieval time by 16ms down to 95ms while six copies only reduce it to 88ms.

It is important to keep in mind that these results are affected by the availability of devices. In a network with lower availability, a producer would need to push more replicas to achieve similar results. On the other extreme, a producer in a highly available network would require fewer replicas to obtain the same level of results.

The results for the network overhead when a different number of replicas are being pushed is presented in Figure 5, for the number of packets and volume of data exchanged.

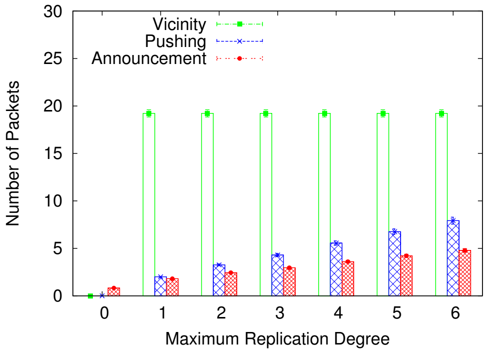


(a) Hit Rate

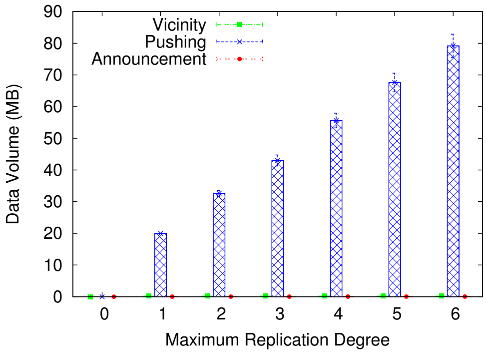


(b) Retrieval Time

Fig. 4. Hit rate and retrieval time results varying the replication degree



(a) Number of Packets



(b) Data Volume

Fig. 5. Network overhead results varying the replication degree

As expected, the more replicas are pushed, the higher the number of packets and volume of data exchanged. The vicinity learning is the dominant cost in terms of packets, as seen in Figure 5(a), despite not varying with different maximum replication degrees (because it is associated with vicinity). Also, when the producer pushes more replicas, it sends more data and creates more providers for the content object, hence increasing the number of announcements. Keep in mind that each announcement forces the routers to update their FIB tables and converge their routing information. As the number of providers increases due to more replicas, this overhead could become significantly detrimental to the network.

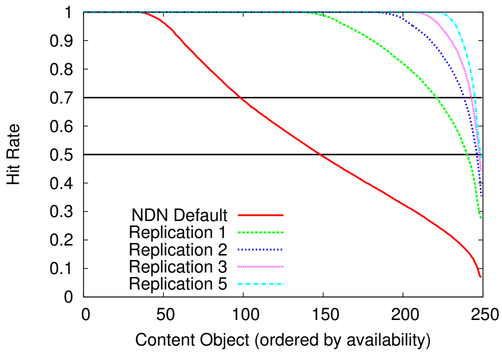
Regarding the data volume the result is straightforward: more replicas lead to more traffic in the network, as seen in Figure 5(b). The data volume added to the network varies from 20MB with one copy until almost 80MB with a maximum of six replicas. In this evaluation, the strategy may push a number of replicas up to the maximum replication degree according to the producer availability, as explained in Section III-C. Since the vicinity and the object size are fixed in 2 and 10MB respectively, one can infer the average number of replicas pushed. For instance, the 80MB data overhead, with the maximum replication degree of six, indicates that only four replicas were pushed in average because of the producer availability. The data volume generated by the vicinity or announcement operations are negligible in comparison to the pushing one.

The conclusion of the replication degree analysis is that pushing a single copy is enough to achieve most of the benefits. Pushing the second copy improves further the producer mobility support. However, pushing three or more copies only increases the results marginally. The network and producer overheads are proportional to the number of replicas pushed, which can be adjusted to find a balance between the benefits obtained and the overhead.

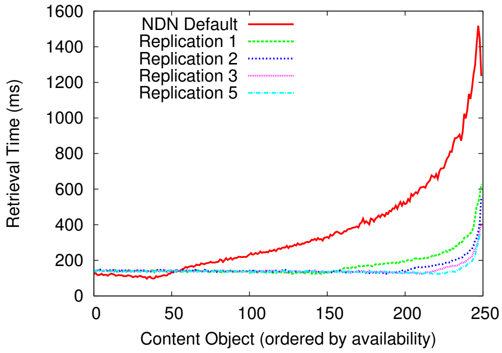
B. *Multiple Producers Scenario*

The second analysis focuses on evaluating the strategy in a scenario with multiple objects produced. Figure 6 shows the results for both hit rate and retrieval time. Throughout the analysis, the objects are classified in terms of availability according to their hit rate, compared to the average device availability of the network (60%). So, we define the following levels according to the hit rates: (a) **low**, for less than 50%; (b) **medium**, for hit rates between 50% and 70%, inclusive, and (c) **high**, for cases above 70%.

With default NDN support to mobility, the content objects have an average hit rate of 61%, as shown in Figure 6(a). The lack of producer mobility support by NDN causes a high variance on the hit rate of content objects, which is determined by the producer availability. Observe that the distribution between high, medium and low availability levels is approximately 40-20-40%, respectively. These results show that the NDN default mechanisms are not enough to provide at least a medium availability (network average availability) for nearly 40% of the objects in the network. Further, the poor availability reflects on the consumers' retrieval time, shown in Figure 6(b). The average time to retrieve a piece of content is 360ms. Despite around 24% of objects being quickly retrieved in between 100 and 150ms, more than 45% of the catalog takes more than 300ms (1.15 times the maximum RTT of the network).



(a) Hit Rate



(b) Retrieval Time

Fig. 6. Hit rate and retrieval time results with multiple content objects

When at least one replica is pushed for each produced content object, the overall hit rate improves significantly to up to 91%. The strategy provides effective support to producer mobility through the replicas. They allow more than 88% of objects (in the worst case) to have a high availability. Although the strategy fails to provide a high hit rate to every object in the system, it limits the low available objects to only 4% in the worst case. The higher hit rate causes consumers to retrieve the objects faster (evidenced by the average retrieval time of 170ms). The percentage of objects that are retrieved in under 150ms increases from 24% to 64%. The fastest retrieval time is around 125ms (25ms higher than with default NDN) due to the announcement convergence time caused by the extra providers in the network. Lastly, the strategy is able to reduce down to just under 6% the percentage of objects that takes more than 300ms to be retrieved.

Although an increase in the number of pushed replicas is beneficial to producers, the gains obtained by each new replica decreases exponentially. Despite pushing more copies marginally improves the results for the average producer as shown in Section V-A2, it allows more producers (especially those with low availability) to achieve better results and highly improve the availability of their content. If the network overhead caused by five replicas can be afforded, the average hit rate rises to 98%, and the number of objects with medium or low availability is reduced to only 2% (or 5 objects). The average retrieval time of objects decreases slightly to 140ms and only 3 objects take more than 300ms to be retrieved.

The network overhead has a similar trend as the one presented in the replication degree evaluation. The vicinity cost has a small variance between producers because of the device distribution in the network. In average, each producer knows about 18 and 21 devices in the vicinity. The pushing and announcement costs, on their turn, grow according to the number of replicas. The overhead is measured during four simulated hours over a network composed of 33 routers. Pushing one replica for each object adds 5860 packets (representing a volume of 5GB). When the strategy pushes up to five replicas, theses values increase to 7738 packets and 17.5GB. The highest overhead measured adds roughly 234 extra packets per router and 1.2MB/s extra traffic in the network.

These results demonstrate that the presented strategy adds homogeneous and equal support to every content object. Even though each producer pushes its objects only using local information, the proposed strategy allows over 88% of objects to remain available under producer mobility. This is a very positive result, considering producers compete for limited network resources without a global coordination, and their local decisions may impact negatively on each other.

VI. CONCLUSION

In this paper, we presented a novel content-centric strategy to support producer mobility in NDN. It leverages key features of NDN to overcome the loss of content availability induced by producer mobility. The key insight is to replicate proactively content by pushing it according to a placement policy. A detailed evaluation of the strategy and its parameters was performed to understand better the trade-offs associated with replicating objects in support to mobility.

We showed that NDN does not support producer mobility adequately, as expected, presenting poor average hit rate (only 60% of requests to objects can be satisfied). Then, we showed the proposed strategy can improve the hit rate by 46% and reduce the retrieval time by 60%. When there are multiple producers, the strategy reduces the percentage of low available content objects from 40% (using only NDN) down to 4%. This benefit is not for free, but the overheads to the network and producers are limited because the strategy restricts the scope and number of replicas through the vicinity size and replication degree.

Regarding future work, we will implement the strategy in NDN and address the re-attachment process. The evaluation of the strategy in a real environment will allow us to gain even more understanding of the strategy.

REFERENCES

[1] C. V. Mobile, "Cisco visual networking index: Global mobile data traffic forecast update, 2014–2019," *San Jose, CA*, 2014.

[2] C. Perkins, "Ip mobility support for ipv4, revised," 2010.

[3] C. Perkins, D. Johnson, and J. Arkko, "Mobility support in ipv6," 2011.

[4] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies*, ser. CoNEXT '09. New York, NY, USA: ACM, 2009, pp. 1–12. [Online]. Available: http://doi.acm.org/10.1145/1658939.1658941

[5] 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. [Online]. Available: http://doi.acm.org/10.1145/2500501

[6] T. Woo, H. Park, S. Jung, and T. Kwon, "Proactive neighbor pushing for enhancing provider mobility support in content-centric networking," in *Ubiquitous and Future Networks (ICUFN), 2014 Sixth International Conf on*, July 2014, pp. 158–163.

[7] J. Lee, S. Cho, and D. Kim, "Device mobility management in content-centric networking," *Communications Magazine, IEEE*, vol. 50, no. 12, pp. 28–34, December 2012.

[8] R. Ying, L. Hongbin, G. Deyun, Z. Huachun, and Z. Hongke, "Lbma: A novel locator based mobility support approach in named data networking," *Communications, China*, vol. 11, no. 4, pp. 111–120, April 2014.

[9] D. Han, M. Lee, K. Cho, T. Kwon, and Y. Choi, "Publisher mobility support in content centric networks," in *Information Networking (ICOIN), 2014 International Conference on*, Feb 2014, pp. 214–219.

[10] R. Ravindran, S. Lo, X. Zhang, and G. Wang, "Supporting seamless mobility in named data networking," in *Communications (ICC), 2012 IEEE International Conference on*, June 2012, pp. 5854–5869.

[11] D.-h. Kim, J.-h. Kim, Y.-s. Kim, H.-s. Yoon, and I. Yeom, "End-to-end mobility support in content centric networks," *International Journal of Communication Systems*, vol. 28, no. 6, pp. 1151–1167, 2015.

[12] J. Tang, H. Zhou, Y. Liu, H. Zhang, and D. Gao, "A source mobility management scheme in content-centric networking," in *Consumer Communications and Networking Conference (CCNC), 2014 IEEE 11th*, Jan 2014, pp. 176–181.

[13] F. Hermans, E. Ngai, and P. Gunningberg, "Global source mobility in the content-centric networking architecture," in *Proceedings of the 1st ACM Workshop on Emerging Name-Oriented Mobile Networking Design - Architecture, Algorithms, and Applications*, ser. NoM '12. New York, NY, USA: ACM, 2012, pp. 13–18. [Online]. Available: http://doi.acm.org/10.1145/2248361.2248366

[14] Y. Luo, J. Eymann, and A. Timm-Giel, "Mobility support for content centric networking," *Telecommunication Systems*, vol. 59, no. 2, pp. 271–288, 2015. [Online]. Available: http://dx.doi.org/10.1007/s11235-014-9960-x

[15] Y. Zhang, H. Zhang, and L. Zhang, "Kite: A mobility support scheme for ndn," in *Proceedings of the 1st International Conference on Information-centric Networking*, ser. ICN '14. New York, NY, USA: ACM, 2014, pp. 179–180. [Online]. Available: http://doi.acm.org/10.1145/2660129.2660159

[16] X. Jiang, J. Bi, and Y. Wang, "What benefits does ndn have in supporting mobility," in *Computers and Communication (ISCC), 2014 IEEE Symposium on*, June 2014, pp. 1–6.

[17] L. Wang, O. Waltari, and J. Kangasharju, "Mobiccn: Mobility support with greedy routing in content-centric networks," in *Global Communications Conference (GLOBECOM), 2013 IEEE*, Dec 2013, pp. 2069–2075.

[18] A. Sharma, X. Tie, H. Uppal, A. Venkataramani, D. Westbrook, and A. Yadav, "A global name service for a highly mobile internetwork," in *Proceedings of the 2014 ACM Conference on SIGCOMM*, ser. SIGCOMM '14. New York, NY, USA: ACM, 2014, pp. 247–258. [Online]. Available: http://doi.acm.org/10.1145/2619239.2626331

[19] J. Ge, S. Wang, Y. Wu, H. Tang, and Y. E, "Performance improvement for source mobility in named data networking based on global–local fib updates," *Peer-to-Peer Networking and Applications*, pp. 1–11, 2015. [Online]. Available: http://dx.doi.org/10.1007/s12083-015-0353-z

[20] Y. Zhang, "User mobility from the view of cellular data networks," in *INFOCOM, 2014 Proceedings IEEE*, April 2014, pp. 1348–1356.

[21] C. Tuduce and T. Gross, "A mobility model based on wlan traces and its validation," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 1, March 2005, pp. 664–674 vol. 1.

[22] J. Tian, J. Hähner, C. Becker, I. Stepanov, and K. Rothermel, "Graph-based mobility model for mobile ad hoc network simulation," in *Simulation Symposium, 2002. Proceedings. 35th Annual*, April 2002, pp. 337–344.

[23] Z. Gao, A. Venkataramani, J. F. Kurose, and S. Heimlicher, "Towards a quantitative comparison of location-independent network architectures," in *Proceedings of the 2014 ACM Conference on SIGCOMM*, ser. SIGCOMM '14. New York, NY, USA: ACM, 2014, pp. 259–270. [Online]. Available: http://doi.acm.org/10.1145/2619239.2626333

[24] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon, "I tube, you tube, everybody tubes: Analyzing the world's largest user generated content video system," in *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC '07. New York, NY, USA: ACM, 2007, pp. 1–14. [Online]. Available: http://doi.acm.org/10.1145/1298306.1298309

[25] F. Zhou, L. Zhang, E. Franco, A. Mislove, R. Revis, and R. Sundaram, "Webcloud: Recruiting social network users to assist in content distribution," in *Network Computing and Applications (NCA), 2012 11th IEEE International Symposium on*, Aug 2012, pp. 10–19.

APPENDIX C — PAPER SUBMITTED TO JNCA

Title: PDRM: A Proactive Data Replication Mechanism to Improve Content Mobility Support in NDN

Journal: Journal of Network and Computer Applications (JNCA)

Type: Research Paper

Qualis: A2

Submission Date: September 27, 2017

Abstract: The problem of handling user mobility has been around since mobile devices became capable of handling multimedia content and is still one of the most relevant challenges in networking. The conventional Internet architecture is inadequate in dealing with an ever-growing number of mobile devices that are both consuming and producing content. Named Data Networking (NDN) is a network architecture that can potentially overcome this mobility challenge. It supports *consumer mobility* by design but fails to offer the same level of support for *content mobility*. Content mobility requires guaranteeing that consumers manage to find and retrieve desired content even when the corresponding producer (or primary host) is not available. In this paper, we propose a proactive replication mechanism that increases content availability through data redundancy in the context of the NDN architecture. Proactive Data Replication Mechanism (PDRM) is a proactive, locality-aware, best-effort, and hint-based replication mechanism that explores available resources from end-users in the vicinity to improve content availability even in the case of producer mobility. We discuss the design of PDRM, evaluate the impact of the number of available providers in the vicinity and in-network cache capacity on its operation, and compare its performance to Vanilla NDN and two state-of-the-art proposals. The evaluation indicates that PDRM improves content mobility support due to using object popularity information and spare resources in the vicinity to help the replication. Results show that PDRM can reduce the download times up to 53.55%, producer load up to 71.6%, inter-domain traffic up to 46.5%, and generated overhead up to 25% in comparison to other mechanisms.

PDRM: A Proactive Data Replication Mechanism to Improve Content Mobility Support in NDN

Matheus B. Lehmann, Lucas Leal, Marinho Barcellos

Institute of Informatics, Federal University of Rio Grande do Sul, Brazil

Andreas Mauthe

School of Computing and Communications, Lancaster University, UK

KOM, TU Darmstadt, Germany

Arun Venkataramani

School of Computer Science, University of Massachusetts, USA

Abstract

The problem of handling user mobility has been around since mobile devices became capable of handling multimedia content and is still one of the most relevant challenges in networking. The conventional Internet architecture is inadequate in dealing with an ever-growing number of mobile devices that are both consuming and producing content. Named Data Networking (NDN) is a network architecture that can potentially overcome this mobility challenge. It supports *consumer mobility* by design but fails to offer the same level of support for *content mobility.* Content mobility requires guaranteeing that consumers manage to find and retrieve desired content even when the corresponding producer (or primary host) is not available. In this paper, we propose a proactive replication mechanism that increases content availability through data redundancy in the context of the NDN architecture. Proactive Data Replication Mechanism (PDRM) is a proactive, locality-aware, best-effort, and hint-based replication mechanism that explores available resources from end-users in the vicinity to

---

Email addresses: `{mblehmann,lsleal,marinho}@inf.ufrgs.br` (Matheus B. Lehmann, Lucas Leal, Marinho Barcellos), `a.mauthe@lancaster.ac.uk` (Andreas Mauthe), `arun@cs.umass.edu` (Arun Venkataramani)

Preprint submitted to Journal of Network and Computer Applications September 27, 2017

improve content availability even in the case of producer mobility. We discuss the design of PDRM, evaluate the impact of the number of available providers in the vicinity and in-network cache capacity on its operation, and compare its performance to Vanilla NDN and two state-of-the-art proposals. The evaluation indicates that PDRM improves content mobility support due to using object popularity information and spare resources in the vicinity to help the replication. Results show that PDRM can reduce the download times up to 53.55%, producer load up to 71.6%, inter-domain traffic up to 46.5%, and generated overhead up to 25% in comparison to other mechanisms.

Keywords: Named Data Networking, Content Mobility, Data Replication, Location Awareness

1. Introduction

Mobile content traffic has been increasing year on year and represents a considerable share of network traffic. At the end of 2016, there were 8 billion mobile devices that generated 7.2 exabytes of traffic per month [1]. These values represent an increase of almost half a billion (429 million) devices and 63% of monthly traffic compared to the previous year [1]. Even with this growing trend, today's Internet architecture fails to provide adequate user mobility support despite mechanisms such as Mobile IPv4 [2] and Mobile IPv6 [3]. The cause is a mismatch between the Internet architecture and the constantly evolving user demands.

Named Data Networking (NDN) [4] is a network architecture proposed to address existing shortcomings of the current Internet, including mobility. Inspired by the predominant user behavior (i.e., the interest of users in content rather than sources or delivery mechanisms), NDN focuses on content, instead of hosts, and builds the network architecture around it. The development of NDN has gained considerable traction with support from industry [5, 6, 7, 8] and its contribution to the work of the IRTF ICN Research Group[1]. Overall, NDN has shown to be a promising solution to handle the growing trend of content dissemination and mobility [1] as well as to be the core component of mobility support in the 5G research and development [9].

The NDN architecture is designed to help the producers to disseminate and keep their content available. Three features stand out to support content mobility: content in NDN is replicable across the network, in-network caching can increase the number of content copies, and any device holding a content copy can satisfy incoming requests for it. As a consequence, content can have additional *providers* besides the producer (or primary host) for its dissemination.

On the one hand, the content-centric design allows the NDN architecture to provide native support for consumer mobility. The communication model employed is receiver-driven and connectionless, which enables consumers to resume content retrieval after moving by just resending their requests seamlessly. This support is simpler and more efficient than MobileIPv6 [3]. On the other, the NDN design is not sufficient to address *content mobility*, offering only limited support and requiring more complex network solutions to achieve this goal [10, 11, 12].

The challenge of content mobility compared to consumer mobility is that objects have to be kept available and reachable for consumers despite a possible movement or unavailability of its providers. With the possibility of multiple providers for the same content in NDN, content is available if there is at least one content provider with a copy of it in the network. Thus, content mobility in NDN is ensured when consumers can continue to retrieve some content despite the unavailability of a fraction of its providers (producer included) due to movement.

In this paper, we propose the Proactive Data Replication Mechanism (PDRM), an enhancement to our previous proposal [13], to improve the content mobility support in NDN. The goal of PDRM is to increase content availability through data redundancy with efficient use of network and end-user resources. PDRM is

---

[1] `https://irtf.org/icnrg`

an optional service for mobile producers to replicate content proactively. Unlike previous approaches, a mobile producer using PDRM learns about its vicinity and uses this information to influence the replication decisions. The learning step is the key feature that allows PDRM to optimize the content replication process for high dissemination performance and low resource consumption.

We evaluate PDRM in three scenarios to quantify its performance and overhead. The first two focus exclusively on the PDRM operation, measuring the impact of the number of available providers and in-network cache capacity on its performance. The third scenario compares PDRM regarding performance and overhead to Vanilla NDN and state-of-the-art approaches, namely Data Spot [14] and Data Depot [15], in the face of producer mobility and clustered content locality.

Experiments indicate that overall PDRM improves content mobility support. The proactive data replication executed by PDRM increases content availability, which consequently reduces the consumer download time, demonstrated in the three scenarios. Regarding the number of available providers, we observe that the download performance improves as the number of providers increases until it converges when the providers saturate. Beyond this point, they can even be detrimental to the content dissemination, reducing the improvement obtained by PDRM. As for the cache capacity, results indicate that PDRM is complementary with in-network caching, resulting in constant performance among all levels of cache size. Lastly, PDRM reduces download times by 53.55%, producer load by 71.6%, inter-domain traffic by 46.5%, and generated overhead by 25% in the comparison scenario. These benefits are indeed the case when producer availability is low, but much more nuanced when the producer is predominantly available.

The rest of the paper is organized as follows. Section 2 defines the concept of content and its mobility support in NDN. Section 3 presents the state-of-the-art regarding content mobility support in NDN. Section 4 discusses the PDRM design to improve content availability. Section 5 describes the methodology used in the evaluation of PDRM regarding proposals, scenarios, metrics, and producer availability. Section 6 presents the results and discusses the key findings of the evaluation. Finally, Section 7 concludes the paper, laying out future work.

2. Background

In this section, we review the background information related to our proposal. The discussion begins with the definition of content and follows with the explanation of the current content mobility support provided by NDN, exposing its shortcomings.

2.1. Content

Content is the key concept of the NDN architecture, which implements it as individual pieces of data called objects. Objects are the basic transfer unit used for the communication between users and are identified with globally unique and hierarchical names. To be transferred, objects are divided into fixed-size chunks and sent as idempotent packets with an embedded cryptographic signature that can be stored and reused in future requests by any device in the network. With these designs choices, chunks, and consequently objects, can be replicable and cacheable in the network because any element with a copy of them has the information needed to guarantee data authenticity and integrity.

The mobility of content depends on the status of existing copies in the network and the devices holding them. Initially, there is only the original content copy generated by the producer (i.e., the original provider). During the content lifetime, events may change the number of content copies and their location in the network. Content availability increases whenever new copies of content are created in three situations: consumers retrieve objects, providers reconnect to the network, or routers cache passing packets. Conversely, content availability decreases when copies are removed due to users deleting stored objects, providers disconnecting from the network, or routers evicting cached packets.

The content mobility challenge is to keep content available (i.e., to have at least one reachable copy of it), minimizing the negative impact of events that reduce content availability on consumers' session continuity. A set of factors may affect positively or negatively the availability and dynamics of the copies. The content popularity impacts the cached content, benefiting popular objects in detriment of unpopular ones. The passing traffic together with the router caching policies may result in more stable or volatile cached copies. Lastly, the user churn impacts the reliability of copies found on end-users.

2.2. NDN Content Mobility Support

Users and routers are the existing entities in an NDN network, and they interact with content in different ways. Users are mobile entities that can act as producers (creating content), providers (serving content data) or consumers (requesting content objects). Due to their inherent mobility, users can disconnect from the network at any given moment, which makes their stored content unavailable for other consumers. Routers, on their turn, are static devices in the network that route INTEREST requests and forward DATA responses. After forwarding a DATA packet back to the consumer, they can cache it to serve future requests.

The NDN architecture provides native in-network caching, which is a data replication feature that could support content mobility by increasing the number of content copies in the network. However, the caching alone proves to be insufficient [10, 11, 12] mainly because of its reactive operation. Since caching operates based on incoming requests, mobility hampers the initial dissemination of content that is unavailable and cannot be found in the network. Given this limitation, NDN requires additional mechanisms to support content mobility properly.

3. State-of-the-Art

In this paper, we investigate content mobility support in NDN focused on improving its availability through content redundancy, known as data rendezvous approach [12]. There are three strategies to increase the number of object copies in the network (and consequently, its availability): Data Caching, Data Spot, and Data Depot. The first one is a reactive process according to the observed traffic whereas the last two strategies are based on replication, which is a proactive operation executed independently (and possibly prior) to consumer requests.

Proactive replication is currently an underexplored approach in NDN, particularly when addressing content mobility. Previous work suggests that proactive replication hardly helps content dissemination and could be replaced by edge caching combined with a simple replacement scheme [16, 17]. However, unlike previous work, we focus on content mobility in ICN, which alters the ratio between replication cost and miss penalty due to content unavailability. Instead of a simple cache miss, consumers may fail to obtain an object from the network. Therefore, a proactive replication approach can be effective under these conditions and worth the investigation.

3.1. Data Caching

The caching strategy is native in the NDN architecture. Routers cache DATA responses that they forward back to consumers and use them to satisfy future requests. The routers decide which DATA packets to store based on caching policies, such as Leave Copy Everywhere (LCE), Leave Copy Down (LCD) [18], and Probabilistic caching [19]. These policies are simple and improve content dissemination (particularly of popular content) without incurring significant processing overhead to routers. The downside of caching regarding mobility is its reactive nature, as it caches content already available. If content is unavailable, it is not served to consumers and, consequently, not cached by routers.

The routers may also employ more sophisticated caching schemes, with a degree of cooperation and collaboration among them [20, 21, 22]. Their overall goal is to reduce the overlapping content in the network caches. Consequently, they improve the overall caching performance by increasing the number of unique objects cached. Although these sophisticated caching schemes provide better performance than simpler ones, they also add higher processing overhead on routers and do not change the reactive nature of caching, which is detrimental when producers are unavailable, and content has not spread in the network yet.

3.2. Data Spot

The Data Spot strategy associates produced content to particular areas in the network where the content is useful, usually around its source location. Authors in [14, 23] propose Data Spot based mechanisms that allow producers to push content to the access router proactively. The proposals are based on two observations: requests are usually routed to the content source location (especially if its availability is uncertain), and the request sequence is predictable because consumers request object chunks sequentially. Before moving, producers offload the content data of current incoming requests to the access router as unsolicited data (i.e., there is no prior request for this data). The access routers are extended to cache instead of discarding the unsolicited data, keeping the content available near its source location to satisfy future requests on behalf of the content producer. The downside of these proposals is adding UNSOLICITED DATA packets, which introduce new vulnerabilities to the architecture and do not guarantee the content availability due to cache volatility.

The mechanism described in [24] is another example of the use of the Data Spot strategy. It proposes a vehicular network based on NDN that disseminates content using a broadcast medium. This characteristic enables any device nearby the sender to store a copy of the transmitted content data, which results in content replication around the source location. The devices storing content can satisfy incoming requests, rebroadcast it to spread further or physically move themselves and the content in the network. The proposals based on ad-hoc networks are hardly generalizable because they are usually domain-restricted and tailored for the target environment. That is, they take advantage of particular characteristics not found in traditional networks, such as the broadcast medium and routing performed by end-users.

3.3. Data Depot

The Data Depot strategy replicates the produced content in fixed servers independent of the source location that serve the received objects on behalf of the providers. In other words, it uses the available resources in data centers instead of those found in users nearby the source location as the Data Spot. This strategy is widely deployed currently in CDN and cloud storage solutions but has been underexplored in the context of NDN. If successfully implemented, Data Depot has the potential of incorporating those existing application-layer solutions to NDN's network layer.

One of the few Data Depot proposals in NDN is a custodian-based solution for content sharing, which takes advantage of the producer's resources across multiple devices [15]. After producing a piece of content, the producer can send it to a trusted device that will become its custodian and may serve it in future requests. The routers are extended to keep a mapping from prefix to custodian to endpoint, allowing them to find the best content copy according to the producer preferences (e.g., prioritize devices with continuous power and connectivity).

4. PDRM

In this section, we extend the work on [13] and propose the Proactive Data Replication Mechanism (PDRM) to improve content mobility support in NDN. PDRM reduces the impact of producer unavailability on the availability of content through data redundancy, with efficient use of available network and user resources. After a producer creates a content object, it can push copies to other users, which become providers. The expected results are an increase in the content retrieval rate and, consequently, an improvement to the consumer's QoE, particularly its download time, with limited overhead for users and the network.

4.1. Overview

PDRM is a proactive, locality-aware, best-effort, and hint-based replication mechanism that explores available resources from end-users in the vicinity. It has two key operations, called vicinity discovery and content push. To proactively deal with mobility, a producer may invoke them on the objects of choice one or more times. First, the producer sends a probe to learn about surrounding devices and the popularity of content in them. Then, the producer decides based on the collected information whether to replicate the objects of choice using the content push operation.

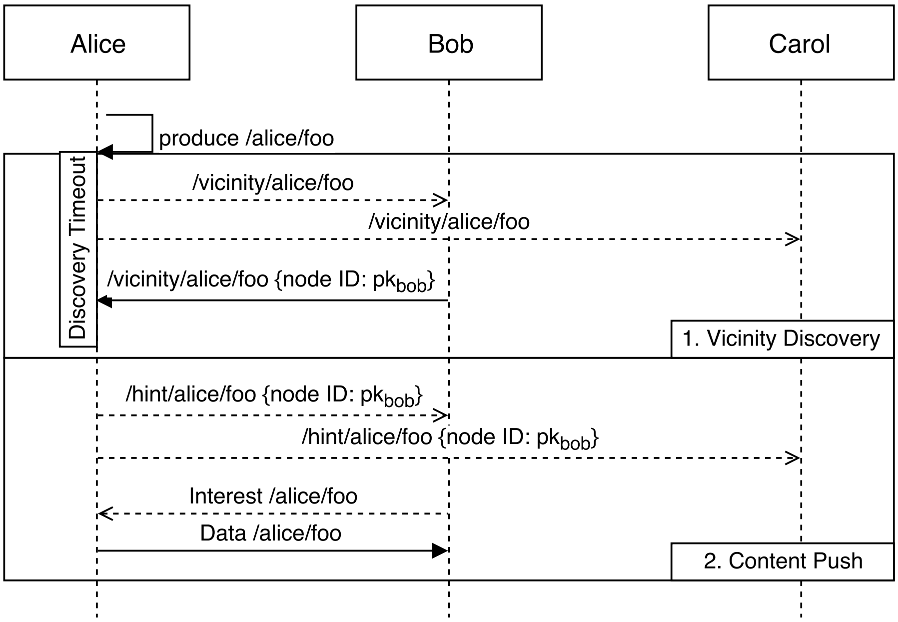


Figure 1: Time diagram of the PDRM operation

The execution of PDRM is illustrated with an example composed of Alice (producer), Bob, and Carol (users in the vicinity of Alice). Figure 1 presents the message exchange of PDRM's execution. This example begins with Alice creating the object named `/alice/foo` and triggering a vicinity discovery for it. If Alice were to create a batch of objects, she could send one vicinity discovery message listing all produced objects and receive a list of interesting objects for each responding user. Alice broadcasts a probe to its vicinity, which contains Bob (interested in the content) and Carol (not interested in the content). Bob replies to the received probe with its public key while Carol just drops the message.

After some period to collect the responses, Alice concludes the vicinity discovery and uses the gathered knowledge to replicate its produced object. In this example, Alice checks that only Bob is interested in becoming a provider for `/alice/foo` and sends him a hint for the content using a broadcast message with his public key. When Bob receives the hint, he proceeds to request the object using the default NDN protocol. Carol also receives the hint but drops it because it is not destined to her.

4.2. Vicinity Discovery

The proactive replication of popular content towards potential consumers improves its availability and dissemination performance. Thus, the first step of PDRM is to obtain relevant context information about the producer location that might help the replication. Mainly, the producer learns about existing devices in the vicinity (i.e., the set of nodes within a distance range that can vary from the direct neighbors to the complete network) and the content popularity in them. The collected information indicates the objects most likely to be requested and their potential consumers.

To discover the vicinity, the producer broadcasts a probe message within a limited scope. As the response, it receives messages from users in the vicinity willing to provide the given content on its behalf. If the producer receives no response, it will not replicate the object due to the lack of interest for the content in the vicinity. PDRM can increase the amount of discovered devices by expanding the vicinity size or enrich the collected knowledge by requesting extra information about users (e.g., their availability).

All the information gathered is used to influence the object replication and copies placement. PDRM enables the producer to collect the consumer identifier (i.e., its public key), availability percentage, and home network location. Because of PDRM's best-effort nature and the unreliability of the data provided by other users, the producer keeps the collected knowledge as a soft state.

PDRM implements its vicinity discovery operation in NDN by extending the NDN INTEREST packet and routers to enable TTL[2], strategy selectors, and long-lived PIT entries as explained next. To trigger the vicinity discovery, the producer broadcasts an INTEREST packet with the extended fields TTL and strategy selectors to the reserved namespace `/vicinity/<object_name>`. The TTL field contains the packet's maximum number of hops, restricting the scope of propagation to the configured vicinity. The strategy selectors, on their turn, define which information the producer is requesting from the potential providers of the content.

The routers are extended to process the reserved namespace `/vicinity` differently from regular INTEREST packets. They create a long-lived PIT entry that is removed only after its expiration period, allowing the requester to receive multiple responses. In comparison, routers consume PIT entries of regular INTEREST packets after they receive an object that satisfies the request. This modification extends the lifetime of PIT entries slightly (i.e., few seconds at most), which should not add new exploits and abuses to the PIT data structure [26, 27, 28].

4.3. Content Push

After the producer builds a view of the consumer interest in its vicinity based on the collected information during the discovery period, PDRM decides how many copies to push (replication degree) and whom to push (placement policy). Further, as NDN does not have a native pushing primitive, PDRM also adds a sender-driven communication operation to enable the producer to push its content proactively to other devices. Next, we discuss the design and implementation of these three aspects: replication degree, placement policy, and push communication.

The increase in content availability obtained by PDRM is related to the replication degree (i.e., the number of copies pushed). The naive approach is to imagine that every object can have multiple copies, making them always

---

[2] NDN does not use TTL currently, but the CCN architecture employs it with the purpose of avoiding loops [25].

available in the network. However, this approach is unrealistic for three reasons. First, proactively pushing content can be expensive, especially for producers with limited resources. Second, the more copies are created, the less effective they become [13]. Third, spare resources from end-users are finite and shared between all producers.

Given these reasons, we expect small replication degrees to be affordable and, even so, not applicable to all objects and producers. PDRM defines the actual number of copies pushed based on the measured content popularity. This design aims at maximizing the benefits of data replication by allocating more resources to objects that have higher popularity and are more likely to be requested in the future. Through the proactive replication, PDRM gives an early start to the dissemination of the selected content, increasing its future availability despite the possibility of the producer becoming unavailable.

After deciding how many copies to create, PDRM selects which users in the vicinity will receive a copy, enabling them to provide the content. PDRM uses the simplest approach as default: picking a random interested user in the vicinity. This policy tends to distribute the load among the set of existing users and avoids overloading a subset of users considered to be the best providers. Nonetheless, PDRM can use different placement policies if there is richer information about the devices nearby. In that case, the producer ranks the user responses according to a set of metrics collected when discovering the vicinity.

Lastly, we discuss the sender-driven communication operation added by PDRM to NDN. It is paramount that such extension does not conflict with NDN's design principles, especially its receiver-driven model. In PDRM, the producer suggests a given content object to other users. A user that receives the indication may follow it or not. In the case of using the hint, the user requests the content and retrieves it from the producer using the NDN protocol. After replicating the object, the producer also relies on the in-network caching to help its content distribution.

PDRM's pushing operation is a hint-scheme implemented at the application layer that uses the NDN receiver-driven communication primitives and complies with NDN design philosophy. Because the producer does not know a name that reaches the selected user, it broadcasts an INTEREST packet containing the targeted user identifier (i.e., its public key) to the reserved namespace `/hint/<object_name>`. The reserved namespace `/hint` does not leave a trail in the PIT because it does not expect a DATA object in response. Instead, upon receiving a hint, if the targeted user accepts the suggestion, it retrieves the hinted object by sending INTEREST packets normally. Otherwise, it justs discards the hint.

4.4. Design Discussion

PDRM has the following characteristics that differentiate it from other state-of-the-art proposals: **proactivity**, **locality-awareness**, **best-effort**, **hint-based**, and **generality**. Next, we discuss each of them.

Proactivity. The proactive approach decouples the replication from the consumer requests, enabling producers to decide when to create copies. With this design, producers can replicate content before they move even if it was not requested, making content available during the mobility period. Consequently, PDRM overcomes the reactive nature of caching, which works well with content that is available or already spread but fails to deal properly with mobile content that is not disseminated yet.

Locality-awareness. Being aware of locality enables PDRM to discover spare resources from other users and the network and use them efficiently. During the vicinity discovery, PDRM gathers information about the current popularity of objects, helping decide which of them to replicate. The leverage of nearby resources and information aims at maximizing the performance gain regarding content availability and download time while consuming the least amount of resources.

Best-effort. The content replication is done in a best-effort fashion and does not require accurate information from the users in the vicinity. Consequently, it can be done under any network condition such as high mobility, dynamicity, or unreliability.

Hint-based. The hinting scheme used by PDRM to push content follows the receiver-driven NDN communication model. It adds a single INTEREST packet per object at the beginning of the transmission to suggest an object to another user. Afterward, the content retrieval is done using the default NDN protocol. Compared to other approaches that alter the NDN architecture and routers significantly, the hinting scheme avoids adding vulnerability issues or processing overhead and still fully benefits from the architectural features (e.g., caching, routing).

Generality. We argue that PDRM is generalizable and can be used in various application domains. In scenarios where the information and resources are costly or unreliable (e.g., IoT and VANET), PDRM can replicate content using limited information of the vicinity and few resources. In conditions with trusted knowledge and abundant resources (e.g., CDN or cloud-storage applications), PDRM can leverage them to provide better content mobility support. Hence, PDRM can benefit a larger set of users and even be incorporated to NDN as a native feature.

5. Methodology

The evaluation of PDRM reported in this paper measures the impact of the number of available providers and in-network cache capacity on the performance of PDRM as well as compares PDRM to Vanilla NDN and the state-of-the-art proposals in a scenario with producer mobility and content clustering. The evaluation was performed by extending the widely popular ndnSIM simulator [29].

5.1. ndnSIM Extensions

ndnSIM [29] is the most complete and realistic NDN simulator available, enabling every aspect of NDN to be simulated. We extended ndnSIM with the implementation of a Mobile Producer (MP) application that uses only the NDN default content mobility support. This case is denoted as *Vanilla NDN* and is the basis for the implementation of PDRM and the two mechanisms based on state of-the-art proposals: *Data Depot* and *Data Spot*. The extensions added to ndnSIM are discussed next[3].

Vanilla NDN. The Vanilla MP relies solely on NDN native in-network caching to support content mobility. It performs three periodic actions: start a session, move, and publish a new object. The first action stops a moving producer and initiates a session at the current location, an active period during which the producer is connected and can provide its content. The second one is the reverse: the producer stops the current session and begins moving. During a movement, the producer is disconnected from the network and does not satisfy consumer requests. The third action produces a new content object and makes it available for interested consumers to request it.

PDRM. The MP with the implementation of the proposed mechanism PDRM, as presented in Section 4. Besides starting a session, moving and, producing content, it can also proactively replicate its content objects in the vicinity with the goal of increasing the number of object copies and, consequently, their availability. The implementation of PDRM extends the INTEREST packet and routers to enable producers to collect information about the vicinity as well as proposes a hint-based content push operation to replicate content.

Data Depot. The Data Depot mechanism is implemented based on the custodian proposal [15]. Its operation is similar to PDRM: the producer pushes the created objects to a selected device in the network, making it the content custodian. The difference between using Data Depot and PDRM is the learning process of potential providers (or custodians). Data Depot has an a priori trusted device at some arbitrary location (e.g., its home network) to which all content is pushed to, similar to a CDN or cloud storage service. In contrast,

---

[3] The source code used in the evaluation is available at `https://github.com/mblehmann/ndnSIM`

PDRM discovers on-demand nearby nodes (in the vicinity) to serve as providers. Hence, the Data Depot mechanism is implemented as a sub-case of PDRM in which producers do not execute the vicinity discovery operation but rather have a previously configured trusted device.

Data Spot. The Data Spot mechanism is based on the proposal that uses unsolicited data [14]. Like PDRM, it pushes content in the network to satisfy future requests. However, Data Spot differs in the content pushing and selection, as explained next. The mechanism forces content to the access router, instead of using hints to send it to other users. For that matter, it extends the DATA packet with an unsolicited data flag that allows it to be sent and cached by routers rather than discarded. Concerning content selection, the mechanism decides what to push based on the content popularity observed from the incoming requests. To obtain this behavior, we extend the MP application to offload the objects expected to be requested during the unavailability period, just before moving.

5.2. Scenarios

We design scenarios to measure the impact of the number of available providers and in-network cache capacity on the performance of PDRM as well as to compare PDRM with state-of-the-art proposals for content mobility. In all scenarios, there is a single producer[4], which follows an on/off model: it becomes unavailable for some non-negligible period, longer than real-time movement. There are multiple consumers which, in their turn, continuously request content that is periodically generated by the producer. Next, we discuss the choice of parameters common to all scenarios. The scenario-specific ones will be presented before the discussion of each result in the next section.

Figure 2 illustrates the network topology used in the evaluation. The routers form a complete binary tree with height 2, representing a content dissemination

---

[4] Note that focusing on a single producer does not affect the generality of results as seen by the similar results obtained using one and multiple producers in our previous work [13].

tree within a domain, similarly to [17]. The producer is connected to the access router, represented by the tree root, and two consumers are connected to each leaf node. Each router in the network can cache up to 1% of the total catalog size [30]. All links have the same capacity, with 30ms delay and 10Gbps bandwidth [31].

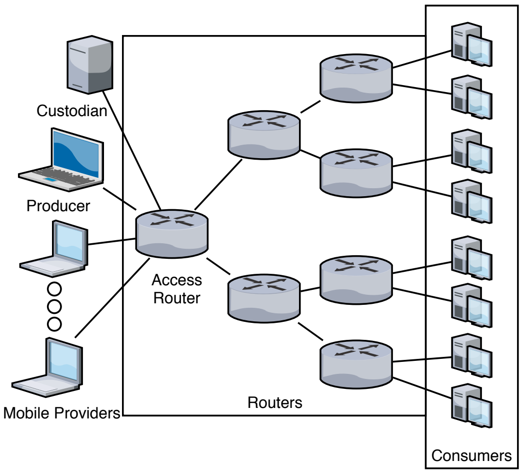


Figure 2: Network topology used in the evaluation

The producer uses one of the content mobility support mechanisms to distribute its content to consumers. Its availability ranges from 30% to 100% in each execution and determines the ratio between session and disconnection times in mobility periods that last 5 minutes. For instance, a producer with 60% availability has succeeding cycles of session and disconnection that last, on average, 180 and 120 seconds respectively. Lastly, the producer also creates a new object periodically every 90 seconds that replaces an existing one from the catalog, keeping its size constant [32].

The catalog has a fixed size of 1000 1MB objects with popularity following a Zipf distribution ($\alpha = 0.8$) [30]. During the simulation, 968K INTEREST requests are issued, divided into 240K in the warm-up period and 728K in the evaluation period [30, 22]. The requests for objects are sent following a Poisson process ($\lambda = 0.02$), resulting in 1 request every 50 seconds. In the case of a failed request, consumers detect a timeout and retransmit it again after the maximum retransmission timeout of ndnSIM, 225 seconds [29]. In summary, 728 objects composed of 1000 1KB chunks each are requested by 8 different consumer streams during the evaluation period, which lasts 75 minutes.

In our evaluation, the ratio between content creation and request is 1.8, which means that few requests happen between the generation of new objects, resulting in a high replication cost. If the ratio increases, the replication cost would be negligible and benefit a larger number of requests. Moreover, NDN through cache and request aggregation would naturally operate better in disseminating content given the popularity distribution and also benefit PDRM. Therefore, we use this ratio in the analysis without loss of generality.

Each of the three mechanisms that extend the Vanilla NDN has different parameters that affect its performance. For object replication, PDRM uses a vicinity size of 2, replication degree of 1, and random placement policy [13]. Additionally, when using PDRM, 20 mobile users with availability ranging from 80% to 100% are connected to the access router. They can become content providers on behalf of the producer but do not send periodic requests like the consumers. Data Depot uses a custodian connected to the access router that stores the last either 700 or 1000 produced objects (70% or 100% of the catalog respectively). Lastly, Data Spot can send up to 10 objects to the access router (its cache size) as unsolicited data.

5.3. Metrics

We analyze both the performance and overhead of each proposal to support content mobility over several runs. The results presented in the next section for each metric collected are the average of all runs. The following metrics have been measured and evaluated.

Object Download Time. The average time elapsed since consumers request the first chunk and retrieve the 1000th (last) chunk of an object. This metric is affected by the content unavailability, which causes timeouts and reissue of pending chunks. The download time represents content availability in the network perceived by the consumers.

Served Data Ratio. The average percentage of chunks served by each existing entity in the network. The result of a chunk request can be either failure (i.e., never retrieved) or success. In the case of success, one of the many potential providers may have served the chunk: producer, routers, or a mechanism-specific element (e.g., providers and custodian).

Extra Traffic Generated. The amount of traffic generated by the MP beyond satisfying INTEREST requests from consumers. That is, it measures the overhead traffic from control and signaling messages (INTEREST packets) as well as pushed objects (DATA packets).

FIB changes. The number of modifications in the routing tables required by each proposal. Some of the proposals rely on announcements on behalf of the producer and dynamic topological adjustments, which causes an overhead to the network concerning routes recalculation and FIB reconfiguration.

6. Evaluation

This section presents and discusses the results obtained from the PDRM evaluation. As mentioned earlier, we investigate a total of three scenarios. The first two focus exclusively on the PDRM operation, measuring the impact of the number of available providers and in network cache capacity on PDRM performance. The last scenario compares PDRM to Vanilla NDN, Data Depot, and Data Spot in the face of varying producer mobility and clustered content.

6.1. Impact of the Number of Available Providers

The first scenario evaluates the impact of the number of potential providers on PDRM because the mechanism relies on them to replicate content. Intuitively, PDRM replication and performance will increase directly with the number of providers available. In this analysis, we measure the effects of the provider pool size on the consumer download time and the fraction of total requests satisfied by the producer, providers, and routers. The results consider a varying number of potential providers in the vicinity between 1 and 20.

Figure 3 presents the average download time of consumers given the number of available providers in the vicinity. There are 5 curves, each representing the producer availability between 30% and 70% with a step of 10%. The results show the impact of both the producer availability and the number of potential providers on the consumer download time.

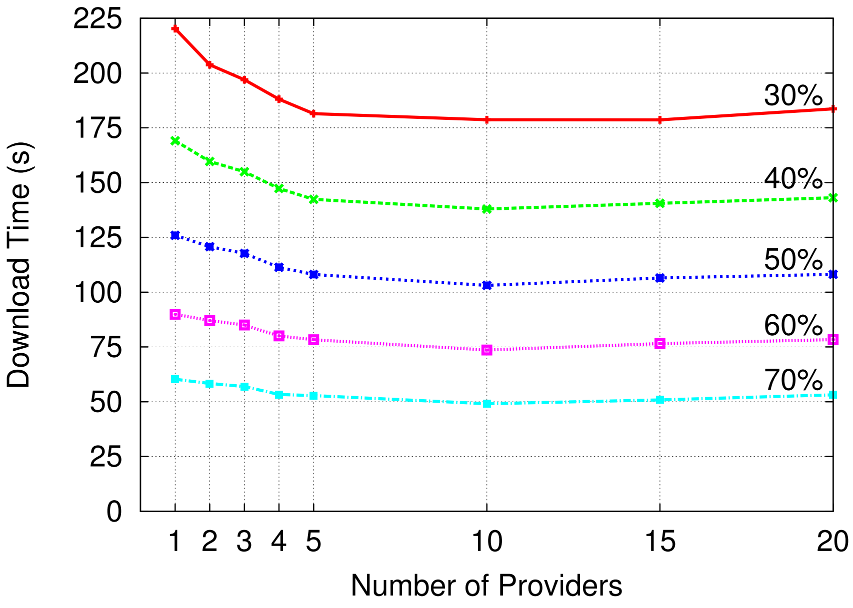


Figure 3: Consumer download time according to the number of potential mobile providers using PDRM

First, by comparing the curves, we observed that the download times decrease as availability increases, as expected. Second, the results show a similar pattern independently of the producer availability. As expected, the longest download times occur when there is a single potential provider in the vicinity besides the producer. As the number of potential providers increases, the download time gradually decreases until it converges at 10 providers, maintaining a similar performance afterward.

The observed behavior indicates a saturation point in which PDRM achieves the lowest download times and having further providers does not improve performance. In this configuration, the ideal number of potential providers is around 10. Having fewer providers diminishes the chance of users willing to provide content, resulting in low replication. In contrast, with larger amounts, the probability of finding providers to receive replicated objects (even unpopular ones) is higher. Moreover, replicating too many objects does not improve the performance significantly while also increases the producer overhead and generates traffic of unpopular content, which can pollute the caches and affect their operation negatively. To minimize this side-effect, PDRM allows a minimum popularity threshold to be set, limiting the replication even when there are interested providers in the vicinity discovered in the discovery period.

We now examine the contribution rate of content served by each network entity, for different numbers of providers, shown in Figure 4. Using PDRM with only 1 provider in the vicinity, the producer serves in average 71.25% of the requests while the remainder is satisfied by the routers (26.21%) and provider (2.54%). This result indicates two benefits of replicating popular content to reduce the producer load: it directly creates new copies on end users and indirectly creates new copies on routers.

We see in Figure 4 that when more potential providers are present, their participation in data distribution increases, varying between 2.53% (1 provider) and 9.75% (20 providers). The low results are a consequence of a low replication percentage of objects in this scenario. The producer and router loads follow the trend discussed during the download time analysis. The producer load decreases while the routers load increases as the number of providers increase until its saturation point. From that point on, the percentage of served content by the routers decreases because of the potential pollution of caches with unpopular objects when replicating a larger amount of content. The replacement of popular objects in the caches reduces their effectiveness and, consequently, increases the producer load, as observed in Figure 4 when the number of providers is higher than 10.

Through the analysis of this scenario, we observe that PDRM requires a certain pool of providers to operate properly. Ideally, the number of potential

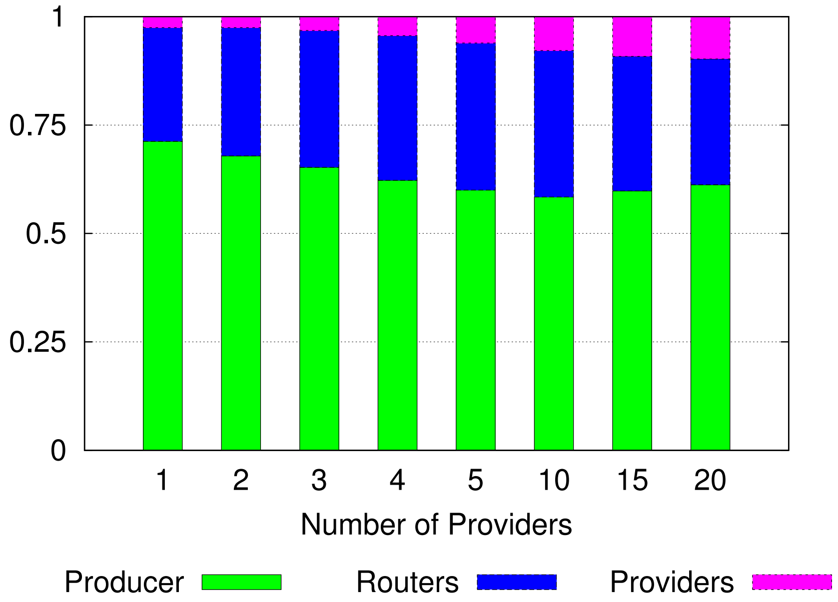


Figure 4: Providing source distribution according to the number of potential mobile providers using PDRM

providers and replication percentage should not be too low or high to achieve the best results. The best values vary according to the scenario configuration and are affected mostly by the content popularity distribution, consumer request rate, and cache capacity.

6.2. Impact of the In-Network Cache Size

The second scenario measures the impact of available in-network caching resources on the content mobility support provided by PDRM. The goal of this analysis is to establish the importance of mobile providers compared to the in-network caching, a major factor for content dissemination in NDN. Similar evaluations in the past have considered cache capacities varying between 0% and 10% of the catalog [30, 22]. We analyze PDRM with cache size values of 0%, 1%, 2%, and 5% [22].

Figure 5 presents the download time of consumers when the producer uses

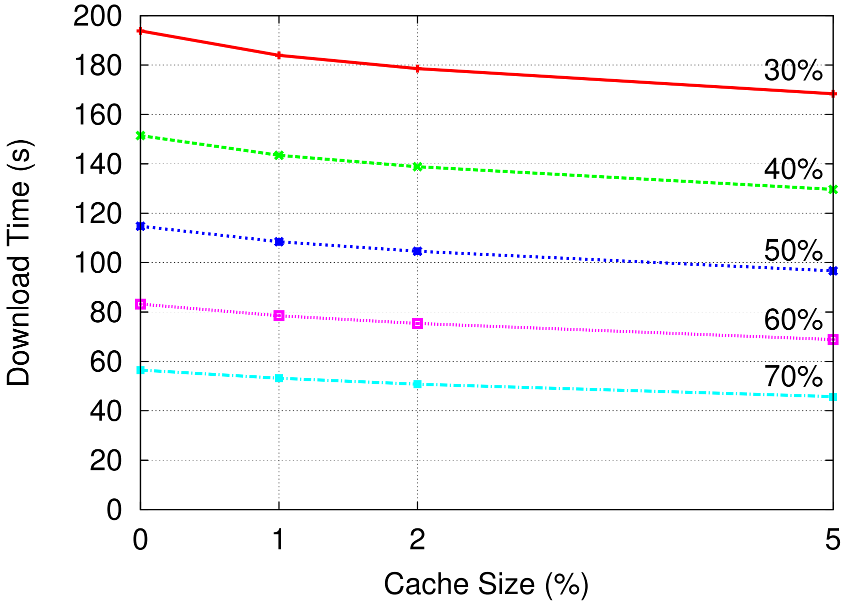


Figure 5: Consumer download time according to the in-network cache sizes using PDRM

PDRM according to four cache sizes. Similar to the previous scenario, there are 5 curves, each representing the producer availability between 30% and 70% with a step of 10%. The results allow an analysis of the impact of both the producer availability and the cache size.

As noted in the previous scenario (§ 6.1), when comparing the curves in Figure 5, we observe that producer availability is the most important factor for download times. Nevertheless, it is well-known that more in-network cache will reduce average download times. We found that caching combined with PDRM achieves a better improvement when the producer has higher availability. In the extreme cases presented in Figure 5, PDRM reduces the download times from 193.80 to 168.34 seconds (30% availability) and from 56.50 to 45.76 seconds (70% availability). The 25.46 seconds reduction in the first case represents a 13.13% proportional decrease while the 10.74 seconds in the second one account for a 19% improvement.

We use Figure 6 to understand further the impact of in-network cache size on PDRM. It shows the fraction of requests served by each element (producer, provider or routers) according to cache size. The results indicate that the providers and routers complement themselves in reducing the producer load of content dissemination. That is, when the routers have less effectiveness (smaller cache sizes), the providers have a bigger role in serving content. When the routers have a higher impact (larger cache sizes), the providers have only a marginal direct contribution disseminating content. In this case, they also have an indirect positive impact by populating the caches with popular content as a side-effect of replication.

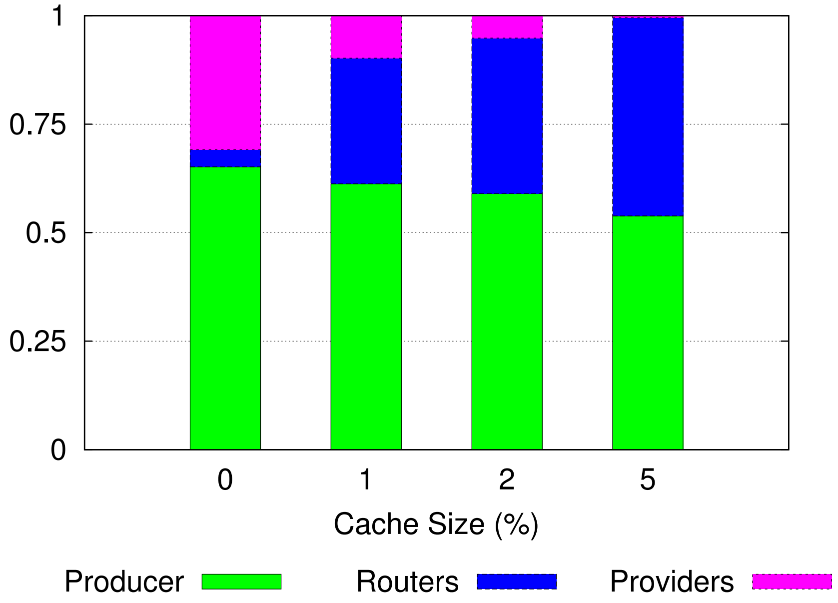


Figure 6: Providing source distribution according to the in-network cache sizes using PDRM

Across all cache configurations, PDRM keeps the producer load between 65.18% (0% cache) and 53.85% (5% cache) on average. With 0% cache, the providers serve 30.84% of the requests while routers contribute with only 3.96%. These values are the consequence of the lack of cache on routers and the increased content availability in the network. Combined, they reduce the effect of request aggregation, a feature in which multiple data responses are distributed with a single interest request.

When caches are added to the network, the providers reduce their direct participation in the content dissemination but increase their indirect impact by populating in-network caches. Numerically, the percentage of content served by them decreases gradually as the cache size increases: 9.77%, 5.17%, and 0.49% respectively for 1%, 2%, and 5% cache sizes. Meanwhile, the routers extend their share in the content dissemination, serving 28.96%, 35.83%, 45.64% of the requests respectively for 1%, 2%, and 5% cache sizes.

This analysis shows that PDRM is a complement for the native in network caching of NDN. The impact of PDRM is inversely related to the amount of cache available in the network, being more significant the lower the cache sizes are. Studies in the ICN literature usually consider small caches (around 1%) [30, 22] as the most representative cases. This configuration results in high competition from objects for the limited cache resources, which benefits PDRM performance greatly.

6.3. Mechanisms Comparison

The third scenario studies the performance of PDRM, Vanilla NDN, and the state-of-the-art mechanisms to support content mobility under producer mobility and content locality. More precisely, consumer requests are determined by their locality in a network composed of multiple domains. The network topology is a complete graph with 8 nodes, each representing a distinct domain, as illustrated in Figure 7. The producer has a home network and moves randomly between the domains. When executing PDRM, there is one potential provider in each domain, totaling 8. In the case of the Data Depot mechanism, the custodian is located in the home network of the producer.

The catalog has 1008 objects, divided into 9 sets containing 112 objects each. They represent the local content of each domain and the global content independent of domains. In each domain, there is a single consumer stream that requests both local and global content in a proportion of 75% and 25% [30]. In addition to the other metrics, we measure the inter-domain traffic, defined by

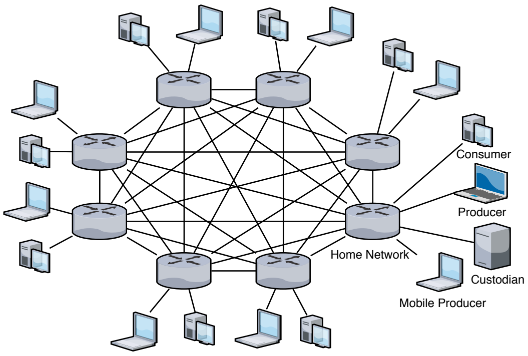


Figure 7: Network topology used for the mechanisms comparison

the amount of traffic flowing between domains.

Figure 8 shows the download times achieved when the producer uses each mechanism. On average, PDRM outperforms Vanilla NDN and Data Spot by 61.09%. In this scenario, Data Spot has a nearly identical performance as Vanilla NDN because its operation is harmed by the object retrieval, which is done almost entirely in a single RTT. Hence, the Data Spot mechanism does not have many opportunities to push pending requests to the routers. PDRM, on the contrary, boosts the download performance especially with lower producer availability because the native caching fails to provide adequate content mobility support. Overall, PDRM takes advantage of content locality in two aspects: it learns about the interested consumers and pushes objects towards them, and it replicates a larger fraction of objects due to the catalog division.

PDRM also achieves similar download performance as Data Depot with a custodian storing up to 70% of the catalog. The average download times across all configurations are 26.424 (PDRM) and 24.572 (Data Depot 70%) seconds. PDRM performs better with lower producer availabilities (less than 55%) but worse with higher ones. This behavior is a consequence of the replication strategies adopted by the mechanisms. PDRM replicates popular objects while Data Depot replicates every content object. This brute force approach may pollute the custodian and caches with unpopular content, reducing their performance.

As expected, PDRM performs significantly worse than Data Depot using a custodian storing 100% of the catalog. This case represents a CDN, in which content is always available on the network, resulting in an average download time of just 3.846 seconds compared to 26.424 seconds obtained using PDRM.

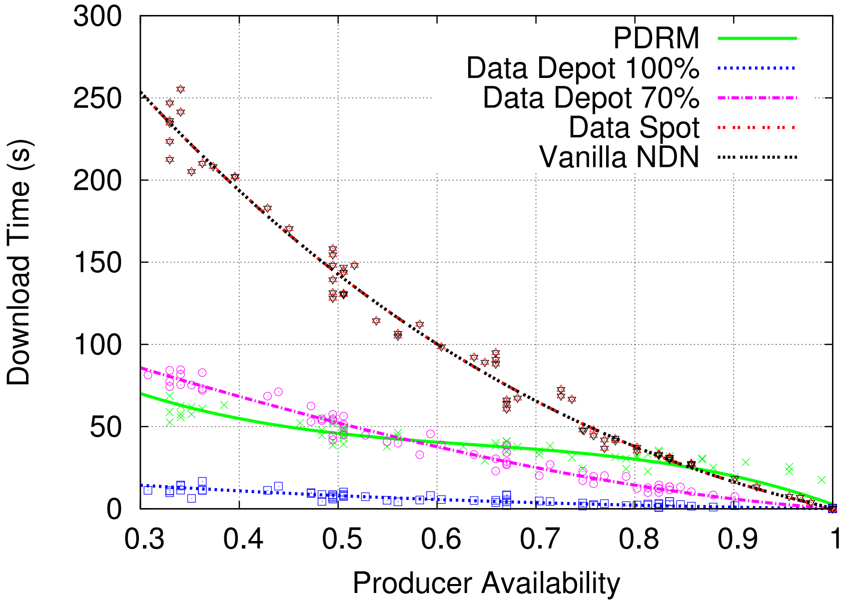


Figure 8: Consumer download time of the mechanisms comparison using PDRM, Data Depot, Data Spot, and Vanilla NDN

Figure 9 shows the providing source distribution achieved by each mechanism and helps to explain the download time results. PDRM reduces the producer load to only 19.64% due to the effectiveness of its proactive replication: the majority of requests is served by providers (75.96%) rather than by routers (4.40%). In summary, PDRM helps the content dissemination by identifying popular content as well as pushing copies to potential providers and towards interested consumers.

Data Depot reduces the producer load even more than PDRM, to 29.90% (70% storage size) and 4.40% (100% storage size). The custodian is always available to serve content and has a storage large enough to store a significant set

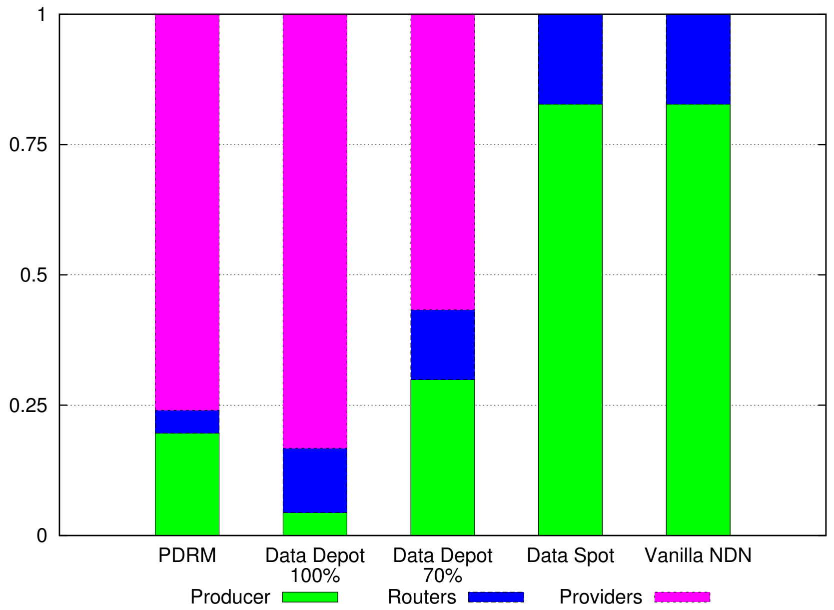


Figure 9: Providing source distribution of the mechanisms comparison using PDRM, Data Depot, Data Spot, and Vanilla NDN

of objects. As a consequence, the custodian satisfies 56.69% and 83.25% of the requests, leaving the remainder 13.41% and 12.35% for the routers, respectively for 70% and 100% storage sizes.

Data Spot and Vanilla NDN reduce the producer load to 82.74%, making routers serve 17.26% of the data. The network topology explains the bad cache efficiency in disseminating content. Each domain has a single level, responsible for both caching and inter-domain routing. Since routers have to forward requests and responses from other domains, they pollute their caches with data not interesting or requested locally.

The overhead of the mechanisms is presented in Table 1. PDRM is the second most consuming mechanism, generating almost 40,000 additional DATA and INTEREST packets while requiring around 39 FIB changes. Compared to Data Depot, the mechanism with the highest overhead, PDRM reduces the number of packets generated and FIB changes in 25% and 60% respectively, independently of the storage size of the custodian. Data Spot pushes fewer copies because of the small-sized objects, reducing the chance of pending requests when starting a movement. Consequently, it generates little overhead.

Proposal	DATA	INTEREST	FIB Changes	External Traffic (% of all traffic)
PDRM	39,466	39,878	39	336.47 MB (41.83%)
Data Depot (70%)	52,450	52,052	104	601.85 MB (78.84%)
Data Depot (100%)	52,450	52,052	104	593.50 MB (77.75%)
Data Spot	400	0	0	581.00 MB (76.11%)
Vanilla NDN	0	0	0	581.05 MB (76.12%)

Table 1: Overhead and data traffic summary of the mechanisms comparison using PDRM, Data Depot, Data Spot, and Vanilla NDN

Finally, we examine the external traffic that each proposal generates, also shown in Table 1. The main goal is to keep traffic within the domain to reduce cost and improve consumer performance. PDRM has the lowest inter-domain traffic among all proposals, with 336MB, representing 41.83% of the total traffic in the network. The explanation for this result is the combination of higher replication percentage and pushing copies towards the consumers. The other three proposals generate around 581MB, which is 76.12% of the total traffic. Noteworthy, Data Depot results in more inter-domain traffic, between 601.85 and 593.50MB, because the producer always sends data to the custodian when replicating.

7. Conclusion

In this paper, we propose the Proactive Data Replication Mechanism (PDRM) to increase content availability through data redundancy maintaining efficient use of network and end-user resources. PDRM differs from other state-of-the-art proposals by learning about its vicinity and using this information to influence the replication decisions. The learning step is the key feature that allows PDRM to optimize the content replication process for high dissemination performance and low resource consumption. We evaluate PDRM the impact of the number of content providers as well as the in-network cache size on its operation and compare its performance to the state-of-the-art proposals for content mobility support.

The conclusions found in the first scenario are that PDRM improves the download performance gradually when adding the first extra content providers until converging the results when the number of providers saturates. At this point, which varies according to the network configuration, PDRM may even fail to achieve its full benefits because of over replication that potentially pollutes the caches with unpopular content. In the second scenario, we show that PDRM works proportionally inverse to in-network caching, providing constant performance independent on the available cache. The comparison scenario evidences the potential of PDRM to support content mobility. Compared to Vanilla NDN, PDRM reduces the download times up to 61.09%, producer load up to 76.26%, and inter-domain traffic up to 46.50%. As expected, PDRM performs worse than Data Depot acting as a CDN. However, PDRM requires no prior storage investment, consumes fewer resources, and may perform better under certain conditions.

Overall, the evaluation indicates promising results for proactive replication in the context of mobility, despite this approach not being suggested in regular networks without mobility [16, 17]. As for future work, there are three extension directions: refinement, evaluation, and deployment. PDRM can be refined to infer better content popularity and how to address it to avoid over replication as well as its economics and how to operate with multiple producers competing for the provider resources. The second direction considers how to evaluate PDRM further: different scenarios, such as real-time or streaming applications, and mobility models (e.g., shorter movement periods). Lastly, PDRM can also be implemented as a prototype and evaluated more accurately in a testbed.

References

[1] C. V. Mobile, Cisco visual networking index: Global mobile data traffic forecast update, 2016–2021, San Jose, CA.

[2] C. E. Perkins, IP Mobility Support for IPv4, Revised, RFC 5944 (Nov. 2010). `doi:10.17487/RFC5944`.
URL `https://rfc-editor.org/rfc/rfc5944.txt`

[3] D. D. B. Johnson, J. Arkko, C. E. Perkins, Mobility Support in IPv6, RFC 6275 (Oct. 2015). `doi:10.17487/rfc6275`.
URL `https://rfc-editor.org/rfc/rfc6275.txt`

[4] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, R. L. Braynard, Networking named content, in: Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies, CoNEXT '09, ACM, New York, NY, USA, 2009, pp. 1–12. `doi:10.1145/1658939.1658941`.
URL `http://doi.acm.org/10.1145/1658939.1658941`

[5] G. Carofiglio, Cisco announces important steps toward adoption of information-centric networking, Blog, https://blogs.cisco.com/sp/cisco-announces-important-steps-toward-adoption-of-information-centric-networking (February 2017).

[6] P. Polakos, Icn: Information centric networking a new networking model for mobile networks and beyond, in: Proceedings of the Workshop on Information Centric Networking for 5G (IC5G 2016), 2016.
URL `https://www.cisco.com/c/dam/m/en_us/service-provider/ciscoknowledgenetwork/files/601_06_29-16-ICN_29Jun2016_CKN_Final.pdf`

[7] I. T. U. (ITU), New itu standard: Information-centric networking to support 5g, Blog, http://newslog.itu.int/archives/1527 (April 2017).

[8] NSF/Intel, Nsf/intel partnership on information-centric networking in wireless edge networks (icn-wen), Program Solicitation, https://www.nsf.gov/pubs/2016/nsf16586/nsf16586.pdf (November 2016).

[9] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. K. Soong, J. C. Zhang, What will 5g be?, IEEE Journal on Selected Areas in Communications 32 (6) (2014) 1065–1082. `doi:10.1109/JSAC.2014.2328098`.

[10] G. Tyson, N. Sastry, R. Cuevas, I. Rimac, A. Mauthe, A survey of mobility in information-centric networks, Commun. ACM 56 (12) (2013) 90–98. `doi:10.1145/2500501`.
URL `http://doi.acm.org/10.1145/2500501`

[11] D. Kutscher, S. Eum, K. Pentikousis, I. Psaras, D. Corujo, D. Saucez, T. C. Schmidt, M. Wählisch, Information-Centric Networking (ICN) Research Challenges, RFC 7927 (Jul. 2016). `doi:10.17487/rfc7927`.
URL `https://rfc-editor.org/rfc/rfc7927.txt`

[12] Y. Zhang, A. Afanasyev, J. Burke, L. Zhang, A survey of mobility support in named data networking, in: 2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2016, pp. 83–88. `doi:10.1109/INFCOMW.2016.7562050`.

[13] M. B. Lehmann, M. P. Barcellos, A. Mauthe, Providing producer mobility support in ndn through proactive data replication, in: NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium, 2016, pp. 383–391. `doi:10.1109/NOMS.2016.7502835`.

[14] T. Woo, H. Park, S. Jung, T. Kwon, Proactive neighbor pushing for enhancing provider mobility support in content-centric networking, in: 2014 Sixth International Conference on Ubiquitous and Future Networks (ICUFN), 2014, pp. 158–163. `doi:10.1109/ICUFN.2014.6876772`.

[15] V. Jacobson, R. L. Braynard, T. Diebert, P. Mahadevan, M. Mosko, N. H. Briggs, S. Barber, M. F. Plass, I. Solis, E. Uzun, B. j. Lee, M. w. Jang, D. Byun, D. K. Smetters, J. D. Thornton, Custodian-based information sharing, IEEE Communications Magazine 50 (7) (2012) 38–43. `doi:10.1109/MCOM.2012.6231277`.

[16] A. Sharma, A. Venkataramani, R. K. Sitaraman, Distributing content simplifies isp traffic engineering, SIGMETRICS Perform. Eval. Rev. 41 (1) (2013) 229–242. `doi:10.1145/2494232.2465764`.
URL `http://doi.acm.org/10.1145/2494232.2465764`

[17] S. K. Fayazbakhsh, Y. Lin, A. Tootoonchian, A. Ghodsi, T. Koponen, B. Maggs, K. Ng, V. Sekar, S. Shenker, Less pain, most of the gain: Incrementally deployable icn, SIGCOMM Comput. Commun. Rev. 43 (4) (2013) 147–158. `doi:10.1145/2534169.2486023`.
URL `http://doi.acm.org/10.1145/2534169.2486023`

[18] N. Laoutaris, H. Che, I. Stavrakakis, The lcd interconnection of lru caches and its analysis, Performance Evaluation 63 (7) (2006) 609 – 634. `doi:http://dx.doi.org/10.1016/j.peva.2005.05.003`.
URL `http://www.sciencedirect.com/science/article/pii/S0166531605000611`

[19] I. Psaras, W. K. Chai, G. Pavlou, Probabilistic in-network caching for information-centric networks, in: Proceedings of the Second Edition of the ICN Workshop on Information-centric Networking, ICN '12, ACM, New York, NY, USA, 2012, pp. 55–60. `doi:10.1145/2342488.2342501`.
URL `http://doi.acm.org/10.1145/2342488.2342501`

[20] S. Saha, A. Lukyanenko, A. Ylä-Jääski, Cooperative caching through routing control in information-centric networks, in: INFOCOM, 2013 Proceedings IEEE, 2013, pp. 100–104. `doi:10.1109/INFCOM.2013.6566743`.

[21] J. M. Wang, J. Zhang, B. Bensaou, Intra-as cooperative caching for content-centric networks, in: Proceedings of the 3rd ACM SIGCOMM Workshop

on Information-centric Networking, ICN '13, ACM, New York, NY, USA, 2013, pp. 61–66. `doi:10.1145/2491224.2491234`.
URL `http://doi.acm.org/10.1145/2491224.2491234`

[22] T. Mick, R. Tourani, S. Misra, Muncc: Multi-hop neighborhood collaborative caching in information centric networks, in: Proceedings of the 3rd ACM Conference on Information-Centric Networking, ACM-ICN '16, ACM, New York, NY, USA, 2016, pp. 93–101. `doi:10.1145/2984356.2984375`.
URL `http://doi.acm.org/10.1145/2984356.2984375`

[23] H. Ko, Y. Kim, D. Suh, S. Pack, A proactive content pushing scheme for provider mobility support in information centric networks, in: 2014 IEEE 11th Consumer Communications and Networking Conference (CCNC), 2014, pp. 523–524. `doi:10.1109/CCNC.2014.6994423`.

[24] G. Grassi, D. Pesavento, G. Pau, R. Vuyyuru, R. Wakikawa, L. Zhang, Vanet via named data networking, in: Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on, 2014, pp. 410–415. `doi:10.1109/INFCOMW.2014.6849267`.

[25] M. Mosko, I. Solis, E. Uzun, C. Wood, Ccnx 1.0 protocol architecture, Tech. rep., Technical report, Aug 2015. url http:// www. ccnx. org/ pubs/ CCNxProtocolArch itecture. pdf (2015).

[26] P. Gasti, G. Tsudik, E. Uzun, L. Zhang, Dos and ddos in named data networking, in: 2013 22nd International Conference on Computer Communication and Networks (ICCCN), 2013, pp. 1–7. `doi:10.1109/ICCCN.2013.6614127`.

[27] M. Virgilio, G. Marchetto, R. Sisto, Pit overload analysis in content centric networks, in: Proceedings of the 3rd ACM SIGCOMM Workshop on Information-centric Networking, ICN '13, ACM, New York, NY, USA, 2013, pp. 67–72. `doi:10.1145/2491224.2491225`.
URL `http://doi.acm.org/10.1145/2491224.2491225`

[28] M. Wählisch, T. C. Schmidt, M. Vahlenkamp, Backscatter from the data plane – threats to stability and security in information-centric network infrastructure, Computer Networks 57 (16) (2013) 3192 – 3206, information Centric Networking. `doi:http://dx.doi.org/10.1016/j.comnet.2013.07.009`.
URL `http://www.sciencedirect.com/science/article/pii/S1389128613002259`

[29] S. Mastorakis, A. Afanasyev, I. Moiseenko, L. Zhang, ndnSIM 2.0: A new version of the NDN simulator for NS-3, Technical Report NDN-0028, NDN (January 2015).

[30] S. Bayhan, L. Wang, J. Ott, J. Kangasharju, A. Sathiaseelan, J. Crowcroft, On content indexing for off-path caching in information-centric networks, in: Proceedings of the 3rd ACM Conference on Information-Centric Networking, ACM-ICN '16, ACM, New York, NY, USA, 2016, pp. 102–111. `doi:10.1145/2984356.2984372`.
URL `http://doi.acm.org/10.1145/2984356.2984372`

[31] J. Garcia-Luna-Aceves, M. Mirzazad-Barijough, E. Hemmati, Content-centric networking at internet scale through the integration of name resolution and routing, in: Proceedings of the 3rd ACM Conference on Information-Centric Networking, ACM-ICN '16, ACM, New York, NY, USA, 2016, pp. 83–92. `doi:10.1145/2984356.2984359`.
URL `http://doi.acm.org/10.1145/2984356.2984359`

[32] S.-E. Elayoubi, J. Roberts, Performance and cost effectiveness of caching in mobile access networks, in: Proceedings of the 2Nd ACM Conference on Information-Centric Networking, ACM-ICN '15, ACM, New York, NY, USA, 2015, pp. 79–88. `doi:10.1145/2810156.2810168`.
URL `http://doi.acm.org/10.1145/2810156.2810168`